

# User Guide to



# SeqNinja®

**DNASTAR, Inc. 2022**

# Table of Contents

<b>Welcome to SeqNinja .....</b>	<b>4</b>
<b>SeqNinja with GUI Help .....</b>	<b>5</b>
Getting Started with SeqNinja .....	6
SeqNinja Tutorials.....	8
Tutorial 1: Inserting bases into an existing sequence .....	9
Part A: Setting up preferences.....	10
Part B: Inserting the bases .....	11
Part C: Running the script .....	13
Tutorial 2: Extracting features as sequences and translating them .....	15
Part A: Defining sequence variables .....	16
Part B: Adding a gene extraction step.....	18
Part C: Translating the extracted genes.....	20
Part D: Extracting and translating features using an alternative method .....	22
Tutorial 3: Reverse complementing and translating a sub-range of a sequence .....	24
Part A: Defining a sub-range of a sequence .....	25
Part B: Adding the reverse complement step .....	27
Part C: Adding the translation step .....	30
Create and Open Projects.....	31
The SeqNinja Interface .....	32
The SeqNinja project .....	33
Welcome screen .....	34
Script views .....	36
Visual view .....	37
Components of the visual view .....	39
Right-click functionality in the Visual view .....	41
Text view .....	43
Console panel.....	45
Toolbar .....	47
Templates panel .....	48
Blank Script.....	50
Add Features.....	51
Change Sequence Origin .....	54
Collect Sequences.....	55
Concatenate Sequences .....	59
Convert File Type .....	61
Create Insertion.....	63
Extract Features as Sequences .....	65
Map Features .....	67
Features screen .....	68

Sequences screen .....	70
Options screen .....	71
Output screen .....	73
Map Features output files .....	75
Feature mapping comparison: MegAlign Pro vs. SeqNinja .....	77
Modify Sequence(s).....	78
Reverse Complement Sequence(s) .....	80
Sample Sequences .....	82
Split Sequences .....	85
Translate .....	87
Write Features.....	89
Toolkit panel .....	91
Settings section .....	92
Converting between annotated and featureless formats .....	95
Sequences section .....	97
Write to Results File .....	98
Define Sequence Variable .....	99
Add Sub-Range.....	100
Add Sequence File .....	101
Add Sequence Text.....	102
Add Sequence Variable.....	103
Functions section .....	104
Annotate .....	105
Collect Sequences .....	106
Cut Sequence(s) .....	107
Extract Features Function .....	108
Reverse Complement.....	109
Sample Sequences .....	110
Strip Metadata.....	111
Translate.....	112
Label as DNA / RNA / Protein .....	113
Label as Circular / Linear .....	114
Other Section .....	115
Add Comment .....	116
Print to Console .....	117
Split Sequence Set.....	118
Define Integer Variable .....	119
Split the SeqNinja screen.....	120
Hide, show and change the sizes of items in the SeqNinja interface .....	121
Work with Sequences.....	122
Add and modify a sequence.....	123
Set sequence ends .....	125

Work with Features .....	127
Work with Scripts .....	133
Specify file paths .....	134
Specify output format and location .....	135
Perform multiple calculations .....	136
Use file patterns.....	137
Use escape codes .....	139
Use setting substitution.....	140
Use editing commands .....	144
Run a script .....	145
Stop running a script.....	148
Access run results .....	149
Use drag & drop .....	152
Save, Close and Exit.....	154
Miscellaneous .....	156
Set preferences .....	157
Default Run Options .....	158
Editor .....	159
Help .....	161
Internet.....	163
Temporary Files .....	164
Turn off usage logging .....	165
Clean cached files on startup.....	166
<b>SeqNinja Command-Line Help .....</b>	<b>167</b>
Getting Started with SeqNinja Command-Line.....	168
Command-Line Options .....	171
The SeqNinja Language.....	172
Escape codes .....	173
File patterns.....	174
Settings .....	175
Integral expressions.....	178
Sequence expressions and assignments .....	179
Sequence filenames, literal sequences and variables .....	180
Multi-sequence files and expressions.....	183
Sequence fragments.....	184
Sequence concatenation.....	187
Sequence functions .....	188
Actions .....	199
Convert Between Annotated and Featureless Formats.....	204
Use Setting Substitution .....	206
Edit in the SeqNinja Shell.....	209
Perform Multiple Calculations.....	210

Example Scripts .....	211
Add segments of literal sequence to every sequence in a set .....	212
Collect sequences with certain attributes into a single file .....	213
Concatenate and de-concatenate contigs for gene finding .....	214
Convert one version of a sequence to another .....	215
Divide a genome into pieces of a specified length & with a specified overlap .....	217
Extract annotated features from a genome .....	218
Extract segments from a sequence .....	220
Extract a specific set of genes or proteins from multiple genomes .....	221
Insert a missing base into a sequence .....	222
Reverse complement a sequence .....	223
Split a multi-sequence file into individual files .....	224
<b>Appendix.....</b>	<b>225</b>
Supported file types .....	226
Change the default application for opening files .....	227
IUPAC codes for amino acids.....	229
IUPAC codes for nucleotides.....	230
Create a script to use as MegAlign Pro input.....	231
Research references.....	233

# Welcome to SeqNinja

---

SeqNinja (also referred to as DNA\*) is a utility for advanced editing of genome sequences and annotations. It supports automation of repetitive tasks via a [custom scripting language](#) that manipulates sets of sequences. This can be useful both for reducing effort, and for repeatability of an operation on updated data.

SeqNinja allows you to edit an annotated sequence and examine the consequence of mutations, facilitating the process of adjusting sequence coordinates between assemblies. Other uses for SeqNinja include file format conversion; batch editing, export and translation of annotations; and merging (concatenation) and splitting of sequence data.

SeqNinja can be run through a graphic user interface (GUI) or through the command line. Each version has its own comprehensive set of help within this User Guide.

- [SeqNinja with GUI Help](#)
- [SeqNinja Command-Line Help](#)

# SeqNinja with GUI Help


---

SeqNinja (also referred to as DNA\*) is a utility for advanced editing of genome sequences and annotations.

SeqNinja allows you to edit an annotated sequence and examine the consequence of mutations, facilitating the process of adjusting sequence coordinates between assemblies. Other uses for SeqNinja include file format conversion; batch editing, export and translation of annotations; and merging (concatenation) and splitting of sequence data.

Each SeqNinja script consists of a user-created script, a list of operations that SeqNinja will execute in the order in which they are written.

- **Beginning** users can easily create a script in the [Visual view](#) by clicking on a pre-defined [template](#) and completing the required fields.
- **Intermediate** users can create a script in the [Visual view](#) by "stacking" [Toolkit panel](#) items.
- **Advanced** users may instead choose to type operations manually into the [Text view](#). The Text view has some functionality that is not covered by the Templates and Toolkit.

Once the script has been created, simply click on the Run button (  ) to [run the script](#). The progress of the run can be monitored in the [Console panel](#). At the conclusion of the run, the output will be saved in the file and format that was specified in the script, and the [Run Results dialog](#) will appear.

# Getting Started with SeqNinja

The following steps demonstrate how to design a new script in the [Visual view](#) using the [Templates panel](#) and [Toolkit panel](#).

1. Launch SeqNinja. Notice that the [Templates panel](#) and [Toolkit panel](#) appear on the right. The [Welcome screen](#) on the left is a launch point to create or open a script, access items in the Templates panel, or access help and tutorials.
2. Choose a template from the Welcome screen or the Templates panel. The [Visual view](#) will open with a copy of the template loaded as a new script.

To do this...	... choose this template:
Create a new, empty SeqNinja (.star) project <a href="#">Blank Script</a>	
Add only specified features to a sequence or feature file	<a href="#">Add Features</a>
Reset the origin of a circular sequences	<a href="#">Change Sequence Origin</a>
Save all files that share a particular naming pattern as a single sequence file or sequence variable	<a href="#">Collect Sequences</a>
Combine two or more sequences into a single, new sequence. (e.g., merge contigs into a pseudo-molecule with a linker)	<a href="#">Concatenate Sequences</a>
Convert a sequence from one type to another (see our list of <a href="#">supported file formats</a> )	<a href="#">Convert File Type</a>
Modify a sequence by inserting new sequence characters	<a href="#">Create Insertion</a>
Extract a set of annotated features from a genome and save them as protein or nucleotide sequences	<a href="#">Extract Features as Sequences</a>
Map specified annotations from a source sequence to a target sequence	<a href="#">Map Features</a>
Create a modified version of a sequence	<a href="#">Modify Sequence</a>
Create the reverse complement of one or more sequences in a file	<a href="#">Reverse Complement Sequence</a>
Create an output file that contains a filtered set of sequences from the source file	<a href="#">Sample Sequences</a>
Save a new, individual output file for each of the sequences in the source file.	<a href="#">Split Sequences</a>
Translate one or more DNA sequences into protein	<a href="#">Translate</a>
Create a SeqNinja feature table or VCF file using the annotations from a specified	<a href="#">Write Features</a>



sequence file	
---------------	--

3. Enter the required information. For example, you may be asked to choose a sequence, specify output format and location, or to select a sequence range.
4. If desired, add additional steps and/or operations to the script using items from the Templates and Toolkit panels. These items can be placed in any order, and some types of operations can be nested inside other operations or steps using [drag & drop](#).
5. Enter the required information for the newly-added steps.
6. If desired, [save the script](#) as a *.star* file.
7. [Run the script](#). SeqNinja will execute the entire script from top to bottom, outputting the results in the file(s) you chose earlier. At the conclusion of the run, the output will be saved in the file(s) and format(s) specified in the script, and the [Run Results dialog](#) will appear. This dialog contains links to the folder(s) containing the output file(s).

# SeqNinja Tutorials

---

Use the links below to navigate to a tutorial of interest. These tutorials are written for the version of SeqNinja with a graphic user interface (GUI). “Part A” of each tutorial contains a link you can use to download the tutorial data.

- [Tutorial 1: Inserting bases into an existing sequence](#)
- [Tutorial 2: Extracting features as sequences and translating them](#)
- [Tutorial 3: Reverse complementing and translating a sub-range of a sequence](#)

In addition to these tutorials, the command-line portion of this SeqNinja User Guide has many [example scripts](#). If you find an example of interest, you can copy the text from its **Script** row and paste it into the [Text view](#) of the SeqNinja GUI. When you switch to the [Visual view](#), it will be populated with the example steps, helping guide you on which templates and options to choose for your own project.

# Tutorial 1: Inserting bases into an existing sequence

---

In this tutorial, you will learn how to insert additional bases into an existing sequence.

Prior to setting up this project, you will also learn how to use the Preferences dialogs to select a working directory for file output, and to specify how SeqNinja should handle “file save” prompts.

Begin with [Part A: Setting up preferences](#).

# Part A: Setting up preferences

To select a “working directory” for file output:

1. Data for all SeqNinja tutorials is located in the same .zip archive. [Click here](#) to download the SeqNinja tutorial data and extract the .zip archive to any convenient location (e.g., your desktop).
2. Launch SeqNinja if it is not already open.
3. Choose **Edit > Preferences** (Win) or **SeqNinja > Preferences** (Mac) and select **Default Run Options** from the list at left.
4. In the Default Run Options dialog, click the **Browse** button and navigate to tutorial data folder extracted in step 1.
5. Click **OK** (Win) or **Open** (Mac) to confirm the selected path.
6. After returning to the Default Run Options dialog, click **Apply**. Do not click **OK** at this point.

The new “working directory” will now be used during script execution as the default location for a `!datapath*` specified with a relative path, and also as the default location for files specified with a relative path, if no appropriate `!datapath*` is set.



**Note:** Setting a preference for a “working directory” does not affect the directory shown by file choosers during the editing of scripts. File choosers instead show the last directory from which a file was chosen.

7. Next, select **Editor** from the list at left.
8. In the Editor dialog locate the checkbox next to **Ask to save before running any script**. By default, this box is checked, causing SeqNinja to prompt you to save each unsaved script before running it. You may choose whether to leave the box checked or unchecked.



**Note:** In subsequent tutorials, you will be asked to save each script manually before running. Therefore, both options will allow you to follow the tutorials as written.

9. Click **Apply** and then **OK**.

Proceed to [Part B: Inserting the bases](#).

## Part B: Inserting the bases

In this part of the tutorial, you will use a template to insert three missing bases (AAA) between positions 24 and 25 of the sequence *pbr322.seq*.

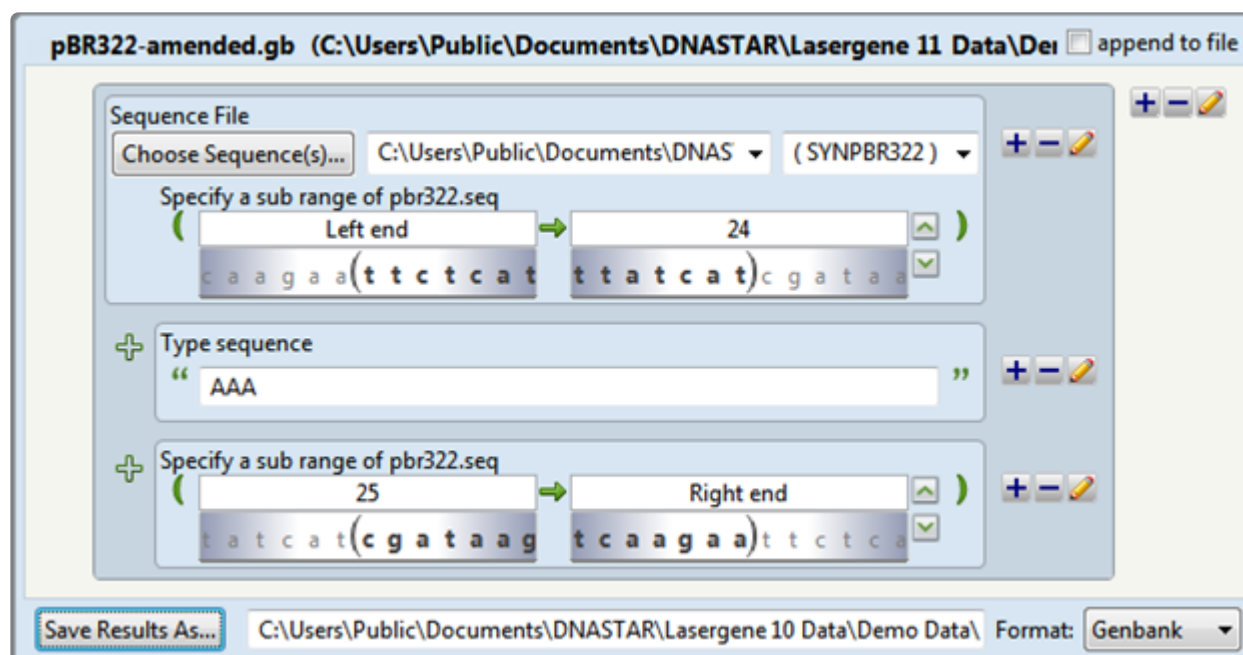
1. Choose **Create Insertion** from the New Script window on the left, or from the Templates panel on the right. Tips for entering your data appear in the yellow “comments” section above the template.

2. In the Sequence File area, press the **Choose Sequence(s)** button.
3. Navigate to the extracted tutorial data folder and double-click on *pbr322.seq*.
4. In the “Type sequence” area, enter the three missing bases. To do this, type over the sample bases in the **Type sequence** row with the bases AAA. You may use small or capital letters but using capitals in this tutorial will allow you to see the changes more easily in the output file.
5. Note the two sets of range boxes above and below the row where you just typed the missing bases.
  - The upper range boxes need to show the sequence segment that occurs *before* the insertion point. In other words, “from the left end of the sequence until base 24.” In the left-hand box,

ensure that one of the following is entered: 1, *Left end*, or *lend* (an abbreviation for “left end”). In the right-hand box, type in the number 24.

- The lower range boxes need to show the sequence segment after the insertion point. In other words, “from 25 until the right end of the sequence.” In the lower pair of range boxes, type in 25 on the left. Acceptable choices for the right box are *Right end* or *rend*. If you knew the length of the sequence, you could also enter that number.
6. To the right of the **Save Results As** button, type in *pbr322-amended*. This output file will automatically be saved in the working directory specified earlier.
  7. In the **Format** drop-down menu, choose the same format as the source file, **Genbank**.
  8. Save the finished SeqNinja script using the **File > Save** command. Navigate to the tutorial data folder (the path is provided in Step 3 of [Part A: Setting up preferences](#)). In the File name box, type *Insertion.star*. Then click **Save**.

The final script (below the comment) should now appear as follows:




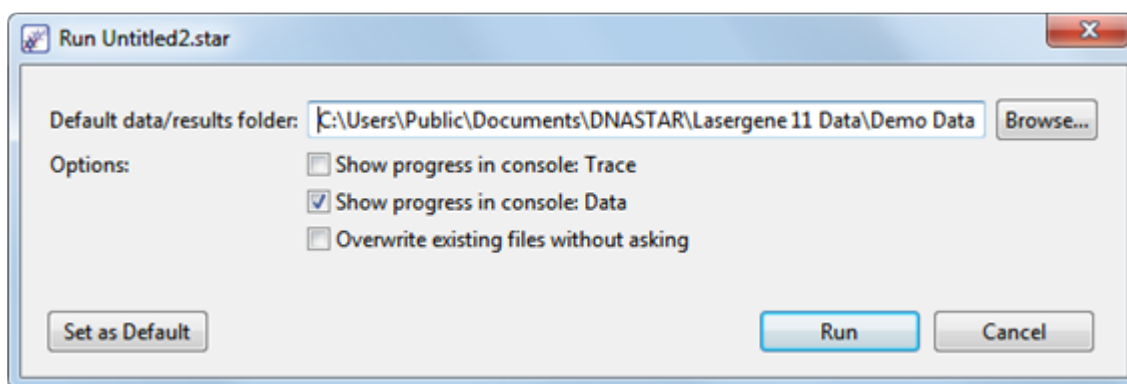
Proceed to [Part C: Running the script](#).


## Part C: Running the script

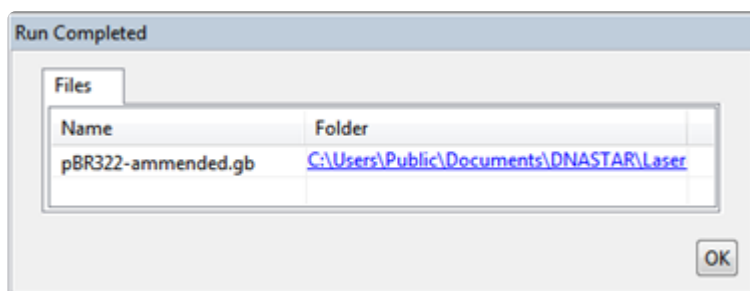
This part of the tutorial will demonstrate several ways to run a SeqNinja script. When you “run” a script, you tell SeqNinja to execute all of the steps in order from top to bottom. The results are then output into the data file and format that you specified in the script.

There are several ways to initiate a run.

- Choosing **Run > Run** from the menu or clicking the Run tool (  ) initiates a run without allowing you to choose any options.
- Selecting **Run > Run with Options** instead launches the following dialog. This dialog allows you to override some of the global preferences from the **Edit > Preferences** dialogs, but for this script only. A useful feature in the Run with Options dialog is the **Overwrite existing files without asking** checkbox. You may check or uncheck this box to hide/show warnings about over-writing an existing data file.



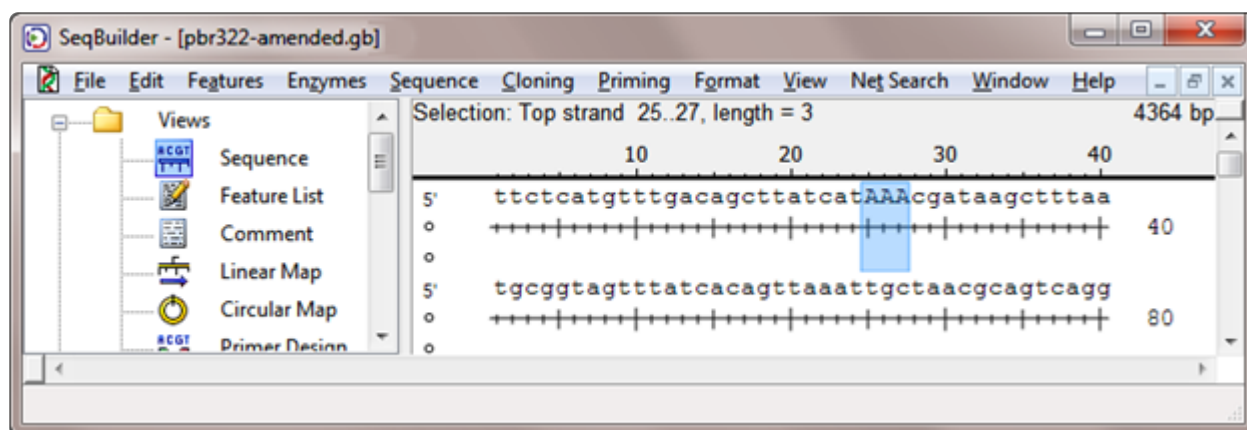
1. For this tutorial, simply press the Run tool (  ). Alternatively, if you opened the Run with Options dialog, keep the settings shown in the image above and click the **Run** button. When the run is complete, the Run Completed dialog is displayed:



If the run was successful, there will be a single tab, called **Files**. The **Files** tab contains a table listing the names of any output files, with links to their directories.

✿ **Note:** A second tab, entitled **Warnings**, may appear after some runs. This tab provides additional information about the run, including warnings about errors (e.g., syntax errors) or about the possibility of unexpected results. For example, if you attempt to reverse complement a protein, the **Warnings** tab will advise you that the sequence being reverse complemented appears to be a protein.

2. In the **Files** tab of the pop-up window, click the link to go to the tutorial data folder.
3. Open the output file *pBR322-amended.gb* in DNASTAR's SeqBuilder Pro application or in any text editor. Note that the three 'A' bases appear, as specified, between base 24 and the base that was formerly base 25.



4. Close SeqBuilder Pro or the text editor and return to SeqNinja.

This is the end of Tutorial 1.



# Tutorial 2: Extracting features as sequences and translating them

---

In SeqNinja, a variable can be used to represent one or more files, variables or functions; or a string of text. In Part A, you will learn how to define a set of three sequence files (*LAMCG.seq*, *pbr322.seq* and *tethis21.seq*) as the variable “A.” Once you have defined this variable, you will use it in a later “gene extraction” step of the script.

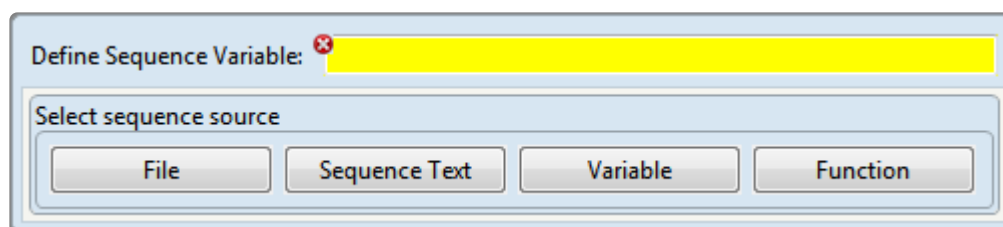


**Note:** The three sequence files used in this tutorial were chosen solely for convenience and are biologically unrelated.

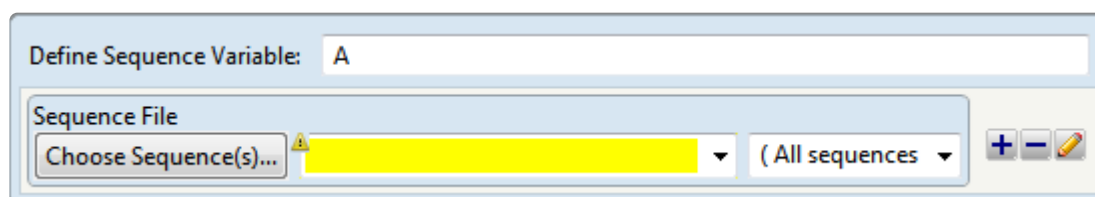
Begin with [Part A: Defining sequence variables and extracting genes](#).


# Part A: Defining sequence variables

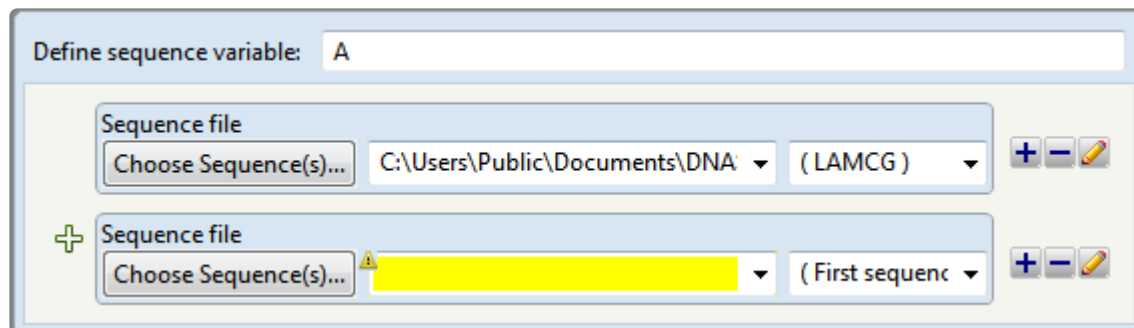
1. Data for all SeqNinja tutorials is located in the same .zip archive. [Click here](#) to download the SeqNinja tutorial data and extract the .zip archive to any convenient location (e.g., your desktop).
2. Launch SeqNinja if it is not already open.
3. From the main window or **Templates** panel, choose **Blank Script**.
4. Click on the **Toolkit** tab on the right of the SeqNinja window to reveal its contents.
5. From the **Sequences** section, select **Define Sequence Variable** to add the following step to the blank script:



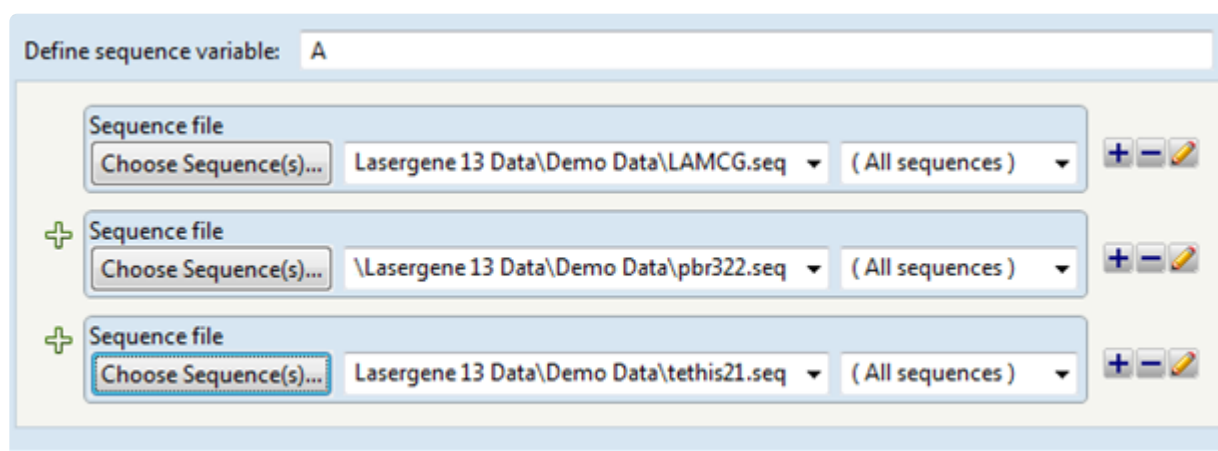
6. Enter the letter *A* in the text box to the right of **Define Sequence Variable**.
7. Since the variable will be defined as being equal to three sequence files, select **File** as the Sequence Source. The lower half of the Define Sequence Variable step changes into a file chooser.



8. To choose the first sequence file, click the **Choose Sequence(s)** button. From the tutorial data folder, double-click on *LAMCG.seq*.
9. To add the next sequence, click the blue plus icon (  ) and select **Add Sequence File**.



10. Press the new **Choose Sequence(s)** button, and then double-click on the next file, *pbr322.seq*.
11. Repeat Steps 8-9 to add the final file, *tethis21.seq*. The Define Sequence Variable step should now appear as follows:



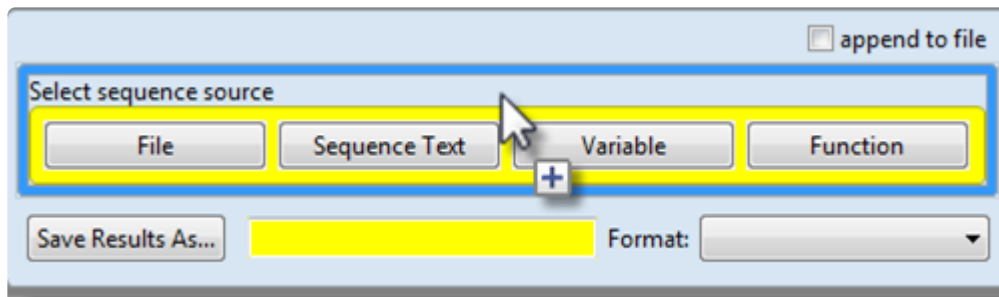
Proceed to [Part B: Adding a gene extraction step](#).

## Part B: Adding a gene extraction step

In this part, you'll add an "extraction" step to the script. This step will extract the genes from the files represented by variable "A" and save them to a single output file.

If you simply dragged the extraction step and dropped it on the Visual View, it would merge with the current step—an undesirable outcome. To make the extraction step appear underneath the current step, you'll actually start by adding a step to save the extraction results (Step 1, below), and only then add the extraction step (Step 2).

1. From the **Toolkit** tab on the right, drag **Write to Results File** (in the Sequences section) and drop it into the Visual View, just below the existing step.
2. Next, drag the **Extract Features** function (in the Functions section) onto the Visual View, and drop it on the Select sequence source area. Drop the function when you see the blue box shown below.



The script now appears as follows:

Define sequence variable: A

Sequence file  
Choose Sequence(s)... .3 Data\Demo Data\LAMCG.seq sequences ) + -

+ Sequence file  
Choose Sequence(s)... 13 Data\Demo Data\pbr322.seq sequences ) + -


+ Sequence file  
Choose Sequence(s)... .3 Data\Demo Data\tethis21.seq sequences ) + -

☐ append to file

**Extract**  
Features: Filter: = + -

Select sequence source  
File Sequence Text Variable Function

Save Results As... Format:

3. In the Extract step, choose **gene** from the **Features** drop-down menu.
4. In the Select sequence source area, choose the **Variable** button. Enter the letter *A* into the Choose sequence variable box.
5. To the right of the **Save Results As** button, type in the filename *Extracted genes*.
6. From the **Format** drop-down menu, choose **Fasta**.
7. Press the **Run** button (  ) to run the script. Click **OK** to close the Run Completed box.

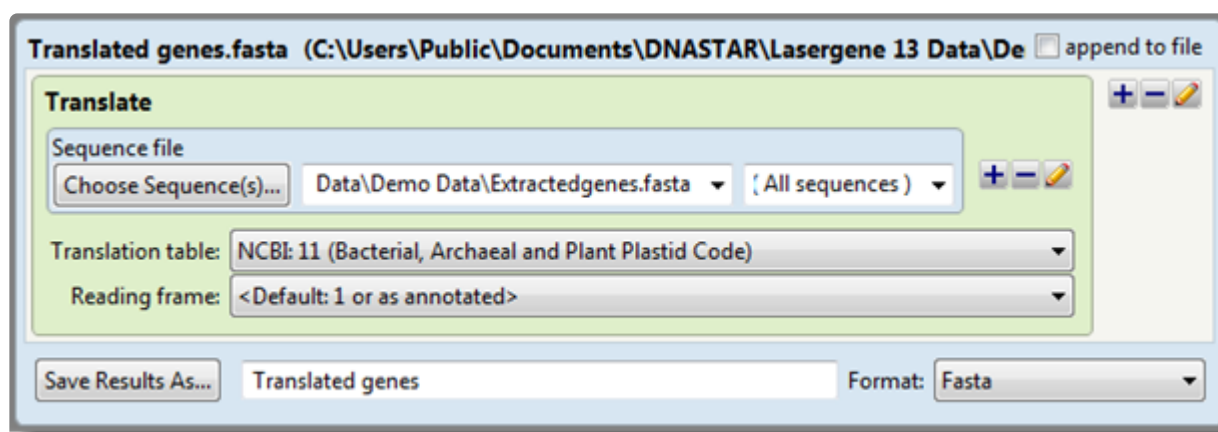
Proceed to [Part C: Translating the extracted genes](#).


## Part C: Translating the extracted genes

In this part of the tutorial, you will translate the output sequence from [Part B](#) and save the results as a second output file.

1. Click the **Templates** tab on the right of the SeqNinja window and choose **Translate**. A new project tab opens with the template loaded.
2. Click the **Choose Sequence(s)** button and navigate to the file *Extracted genes.fasta*.
3. From the **Translation Table** drop-down menu, choose **NCBI: 11**. Leave the **Reading Frame** drop-down menu at the default setting.
4. To the right of the **Save Results As** button, type the name *Translated genes*. From the **Format** drop-down menu, choose **Fasta**. The output file will automatically be saved in the working directory specified earlier.

The script now appears as follows:



5. Run the script by clicking the **Run** tool (  **Run** ). When the run finishes, the Run Completed dialog will appear. Click on the link to open the tutorial data folder.
6. To view the translation results, open *Translated genes.fasta* in DNASTAR's Protean 3D or in any text editor. The table below shows the extracted and translated genes from the three files.

Filename	Extracted and translated genes
LAMCG.seq	<i>int</i>
	<i>exo</i>
pbr322.seq	<i>tet</i>

	<i>bla</i>
<i>tethis21.seq</i>	(none found)

As you can see from the table, there were no annotated genes in the *tethis21.seq* file. However, all three files do have annotated CDS features. To extract and translate these:

7. Go back to the SeqNinja project tab that contains the extraction step and select **CDS** from the **Features** drop-down menu . Click **Run** again and overwrite the previous result file.
8. Click on the SeqNinja project tab with the translation step and click **Run**. Overwrite the previous files. Open the results in a text editor and see how they differ from those in Step 6.

Proceed to [Part D: Extracting and translating features using an alternative method](#).

## Part D: Extracting and translating features using an alternative method

---

When translating a *.gbk* or *.seq* file that contains CDS features, the extraction step in Part B is redundant. The reason is that, during translation, SeqNinja automatically translates CDS features rather than the entire sequence. To see a different method of extracting CDS features than was used in [Part A](#) and [Part B](#):

1. Open the SeqNinja Text View by clicking the **Text** tab near the bottom left corner of the SeqNinja window or by choosing **View > Text View**.

The Text view contains the same information as the Visual View, but in a more concise format. With experience, you may be able to write a script more quickly in the Text View than the Visual View. The Text View is also more efficient than the Visual View when adding large amounts of data, such as lists of sequence sub-ranges, or when copying and pasting, as in our next step.

2. Overwrite the existing text by copying and pasting the text below:

```
$A = "LAMCG.seq"+"pbr322.seq"+"tethis21.seq" "Translatedgenes.fasta" =  
translate($A,'/transl_table=11')
```

3. To view the revised script in the Visual view, select the **Visual** tab or use the **View > Visual View** command.



Define sequence variable: A

Sequence file		
Choose Sequence(s)...	LAMCG.seq	( All sequences )
+ Choose Sequence(s)...	pbr322.seq	( All sequences )
+ Choose Sequence(s)...	tethis21.seq	( All sequences )

**Translatedgenes.fasta** (C:\Users\Public\Documents\DNASTAR\Lasergene 13 Data\Der ☐ append to file

**Translate**

Choose sequence variable  
\$ A

Translation table: NCBI: 11 (Bacterial, Archaeal and Plant Plastid Code)

Reading frame: <Default: 1 or as annotated>

Save Results As... Translatedgenes.fasta Format: Fasta

Note that the extraction step has been eliminated, and the sequence variable is now utilized in the translation step.

This is the end of Tutorial 2.

# Tutorial 3: Reverse complementing and translating a sub-range of a sequence


---

This tutorial explores how to reverse complement and then translate a specified sub-range of a sequence. In this example, you will save the translated reverse complement of the int gene and seven codons on either side of it from the file *LAMCG.seq*.

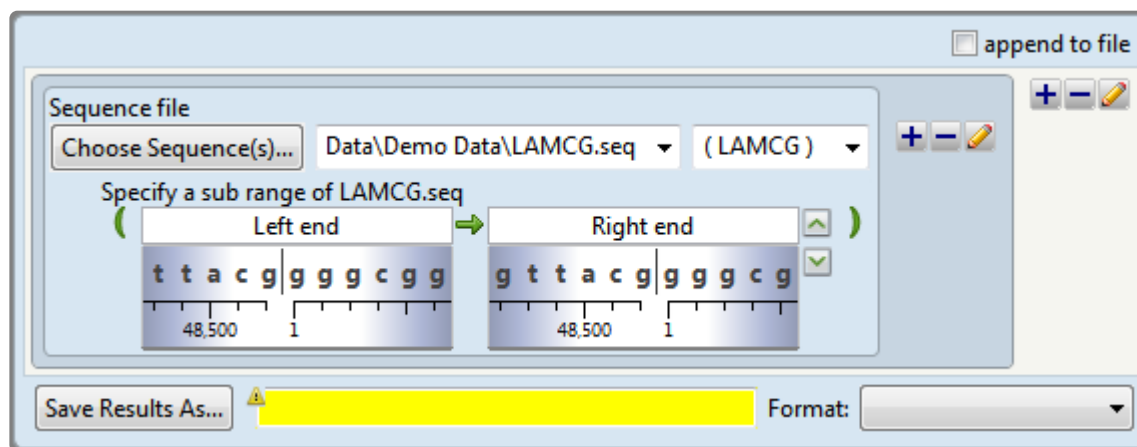
Sub-ranges can easily be set using the **Modify Sequence(s)** template. However, this tutorial demonstrates how to set up the same steps using the **Toolkit** only.

Begin with [Part A: Defining a sub-range of a sequence](#).

## Part A: Defining a sub-range of a sequence

1. Data for all SeqNinja tutorials is located in the same .zip archive. [Click here](#) to download the SeqNinja tutorial data and extract the .zip archive to any convenient location (e.g., your desktop).
2. Launch SeqNinja if it is not already open.
3. From the main window or **Templates** panel, choose **Blank Script**.
4. Open the **Toolkit** panel and click **Write to Results File** (in the Sequences area) to add it to the Visual View.
5. Since the sequence source is a file, click the **File** button. Press the **Choose Sequence(s)** button and double-click on the tutorial data file *LAMCG.seq* to select it as the source sequence.
6. In order to add a step for setting the ends of the sequence, do any of the following:
  - Right-click in the blue “Sequence file” area of the existing step and choose **Add Sub-Range**.
  - Click on the plus icon (  ) to the right of the **Choose Sequences** row and select **Add Sub-Range**.
  - Return to the **Toolkit** panel and click **Add Sub-Range** from the Sequences section.

The script now appears as follows:



Two shaded “wheels” appear under the range boxes showing the sequence surrounding each coordinate. Note the similarities between the sub-range boxes and the double set of range boxes in Tutorial 1: Inserting Bases into an Existing Sequence.

7. The *int* gene occupies a range from 27812 to 28882 bp. To scroll to the gene, type over the words *Left end* with 27812 and the words *Right end* with 28882. Observe that the nucleotides and ruler

under the text boxes change to reflect the new positions.



8. To view the sequence annotations for this region, click the downward-pointing green arrow on the right (▼). The int gene graphic, which appears as two large arrows, verifies that the end coordinates do indeed bracket the correct gene.

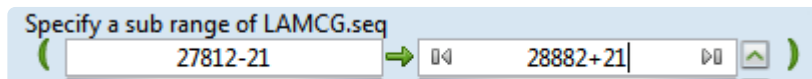
The screenshot displays the SeqNinja software interface for viewing sequence annotations. The main window is titled "Sequence file" and shows a dropdown menu for "ergene 15 Data\Demo Data\LAMCG.seq". Below this, there is a "Specify a sub range of LAMCG.seq" section with input fields for coordinates. The left panel shows coordinates 27,810 to 27,812, and the right panel shows 28,880 to 28,882. Both panels display a sequence view with coordinates and a corresponding gene annotation view. The left panel shows mRNA-pl (alt; via t'j4 terminator) and mRNA-pl (alt; via t'j3 terminator) annotations. The right panel shows mRNA-pl (alt; via t'j4 terminator), mRNA-pl (alt; via t'j3 terminator), mRNA-pl (alt; via t'j2 terminator), and mRNA-pl (alt; via t'j1 terminator) annotations. The gene annotation view shows two large arrows labeled "int" (integration) and a blue arrow labeled "AAA9...". The interface includes a "Save Results As..." button at the bottom left and a "Format:" dropdown menu at the bottom right.

Proceed to [Part B: Adding the reverse complement step](#).

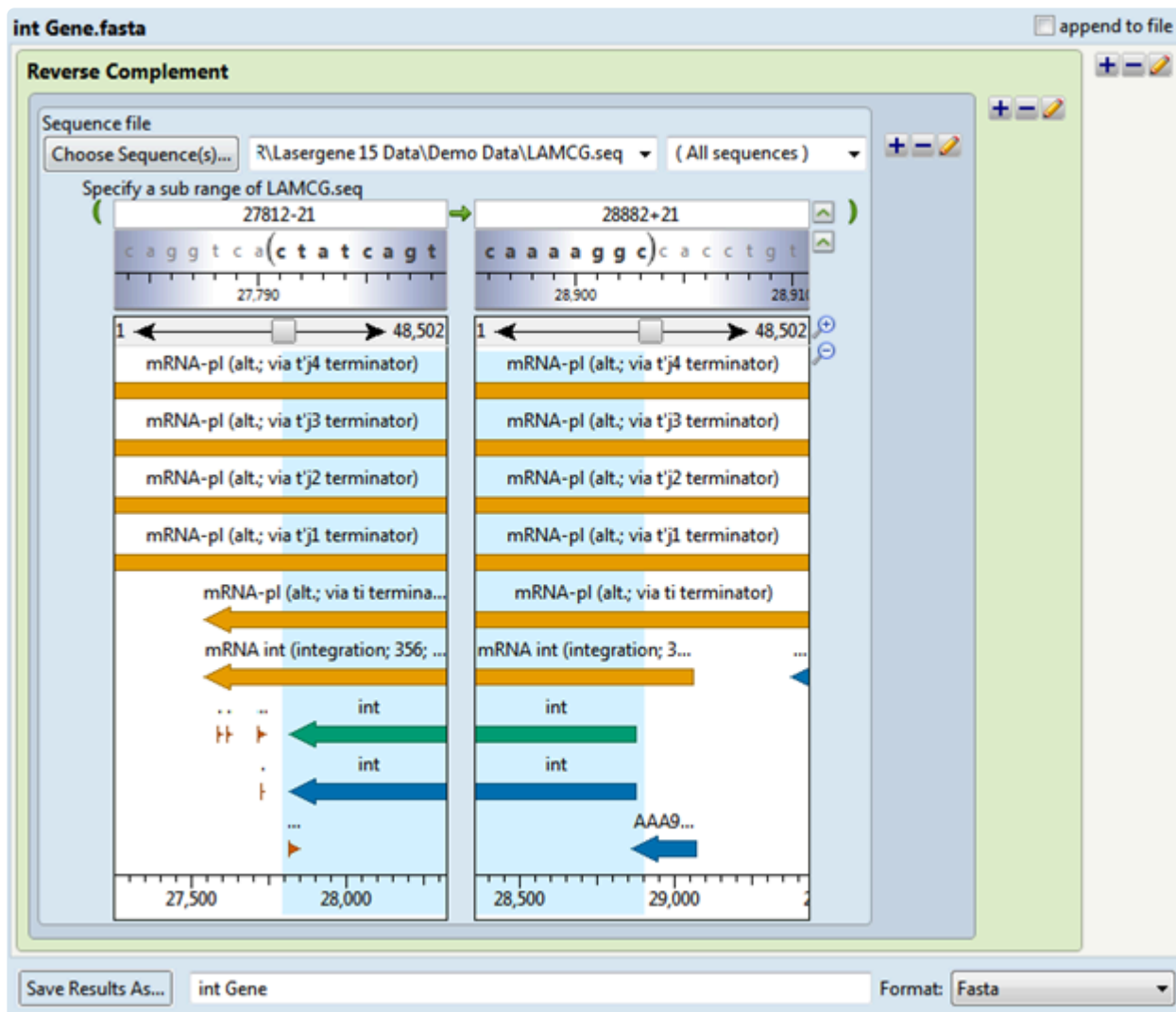
## Part B: Adding the reverse complement step


Observe that the arrows for the *int* gene are pointing left, indicating that it is on the opposite strand. Because we will be dealing with the isolated gene sequence, we will take the reverse complement of the sequence.

1. Click on the pencil icon () in the outer box (white background) and select **Reverse Complement**. The features display will close. Re-open the features display by clicking on the green arrow () , as in the previous step.
2. Assume you want to save not only the gene, but also seven codons on either side of it. Subtract 21 from the value on the left by typing “-21” after the existing coordinate. Add 21 to the value on the right by typing “+21” after that coordinate. That is,  $27812 - 21$  in the left box and  $28882 + 21$  in the right box. Note that since SeqNinja will accept integer expressions as coordinates, it is not necessary to do the calculations and enter the values 27791 and 28903. However, that would work equally well.



3. To the right of **Save Results As**, type the name *int Gene*.
4. From the **Format** drop-down menu, choose **Fasta**. The script now appears as shown:




- Click the **Run** button (  **Run** ).
- If you now open the results file *int Gene.fasta* in DNASTAR's SeqBuilder Pro or a suitable text editor, it will appear as follows:

```
>LAMCG-rc COMMENT      Sequence 'LAMCG-rc' generated by SeqNinja on Feb 25, 2016 at 12:06:06 PM CST.
gccttttgaagaggatcagaaatgggaagaaggcgaagtcagcgcgggatttacc
cctaaccctttatataagaacaatggatattactgctacagggaaccaaggacgggtaaa
gagtttggattaggcagagacaggcgaatcgcaatcactgaagctatacaggccaacatt
gagttattttcaggacacacacacaaagcctctgacagcgagaatcaacagtgataattcc
gttacgttacattcatggcttgatcgctacgaaaaaatcctggccagcagaggaatcaag
cagaagacactcataaattacatgagcaaaattaaagcaataaggaggggtctgcctgat
gctccacttgaagacatcaccacaaaaaagaattgoggcaatgctcaatggatacatagac
gagggaaggcggtcgctcagccaagttaatcagatcaacactgagcgatgcattccgagag
gcaatagctgaaggccatataacaacaaaccatgtcgctgccactcgcgcagcaaaatca
gaggtgaaggagatcaagacttacggctgacgaatacctgaaaaatttatcaagcagcagaa
tcattaccatgttggctcagacttgcaatggaactggctgtgttacccgggcaacaggtt
ggtgatttatgcgaatgaagtggctgatatcgtagatggatattctttatgtcgagcaa
agcaaaacaggcgttaaaattgccatcccaacagcattgcattgtatgctctcggaata
tcaatgaaggaaacacttgataaatgcaaagagattcttggcgggagaaaccataattgca
tctactcgtcgcgaaccgcttccatccggcacagtatcaaggtattttatgcgcgcacga
aaagcatcaggtctttccttcgaaggggatccgcctacctttcacgagttgcgagtttg
tctgcaagactctatgagaagcagataaagcgataagtttgctcaacatcttctcgggcat
aagtcggacaccatggcatcacagtatcgtgatgacagaggcaggagtgaggacaaaatt
gaaatcaaataatgatattttttgactgatag
```

Proceed to [Part C: Adding the translation step](#).

## Part C: Adding the translation step

To add a translation step after the reverse complement step:

1. Click on the pencil icon () in the outer box (white background) and select \*Translate. The translation step is added and encompasses the reverse complement step and the source sequence.
2. Use the **Translation table** drop-down menu to select **NCBI: 11 (Bacterial, Archaeal and Plant Plastid Code)**.
3. In the Save **Results As** textbox, change the name to *translated int Gene*. Click **Run**.
4. Open the output file in SeqBuilder Pro or in a suitable text editor. It should appear as below.

```
>int-transl COMMENT      Sequence 'int-transl' generated by SeqNinja on Oct 24, 2018 at
2:33:07 PM CDT.
mrrrrsherrdlppnlyirnnngyycyrdprtgkefqlgrdrriaiteaiaqanielfsgkh
hkpltarinsdsvtlhswldryekilasrgikqktlinymkikairrglpdapledit
tkeiaamngyidegkaasaklirstlsdafreaiaeghittnhvaatraaksevrssrl
tadeylkiyqaaesspcwlrlamelavvtgqrvgdicemkwsdivdgylyveqsktgkvi
aiptalhidalgismketldkckeilggetiiastrreplssgtvsryfmrarkasglf
egdpptfhelrslsarlyekqisdckfaqhllghksdtmasqyrddrgrewdkieik.
>AAA96563.1-transl COMMENT      Sequence 'AAA96563.1-transl' generated by SeqNinja on Oct 24,
2018 at 2:33:08 PM CDT.
af.rgsemgrrrsh
```

When SeqNinja translates a feature from a sub-region of a GenBank file, or a .seq file with GenBank style annotations, any overlapping feature (or portion of a feature) is also translated if the reading frame is not specified. The second sequence below is a translation of a piece of the gene *xis* that overlaps *int* and is translated in frame 1 instead of the native frame 3 used for *xis*.

This is the end of Tutorial 3.



# Create and Open Projects

---

To create a new SeqNinja *.star* file:

Do any of the following:

- From the [Welcome screen](#), press **New Script** on the left and choose **New blank script** on the right.
- Choose **File > New Script** or press **Ctrl/Cmd+N**.
- Choose **Window > New Window**.

To open an existing SeqNinja *.star* file:

Do any of the following:

- From the [Welcome screen](#), press **Open Script** on the left. On the right, choose **Open** or choose a file from the **Recent documents** list.
- Choose **File > Open**.
- Press **Ctrl/Cmd+O**.
- Choose **File > Recent Documents** and choose a file from the list.
- Choose **Window** and choose a file from the list.

# The SeqNinja Interface

When you first launch SeqNinja, the [Templates panel](#) and [Toolkit panel](#) appear on the right. The [Visual view](#) on the left initially displays the SeqNinja Template Gallery, a list of shortcuts to items in the Templates panel.



**Note:** The SeqNinja Template Gallery also appears when you choose the **File > New Script** command.

If you plan to use the Visual View, simply click on a shortcut to begin your project. If you will be using the [Text view](#), instead click the **Text** tab at the bottom of the window to get started.

The components of the SeqNinja user interface include:

Component	Description
<b>Scripts</b>	Multiple SeqNinja scripts can be open at the same time. Each script is represented by a tab above the Script view. To make a script active, click its tab. To close a script, click the 'x' on its tab.
<b>Menus</b>	SeqNinja has six menus: <b>File</b> , <b>Edit</b> , <b>View</b> , <b>Run</b> , <b>Window</b> and <b>Help</b> .
<b>Script View</b>	Displays your choice of either a <a href="#">Visual view</a> or <a href="#">Text view</a> . Select the desired view by clicking one of the view tabs at the bottom left of the script. Both views contain the same information in different formats.
<b>Console panel</b>	The <a href="#">Console panel</a> displays the output of the script.
<b>Toolbar</b>	The <a href="#">toolbar</a> contains items that can be used in one or more views. The <a href="#">Visual view</a> and <a href="#">Console panel</a> also have their own sets of tools.
<b>Two docking areas</b>	<p>These initially contain two docked <b>panels</b> and four <b>expand bars</b>:</p> <ul style="list-style-type: none"> <li>• The <a href="#">Templates panel</a>.</li> <li>• The <a href="#">Toolkit panel</a>, including the <a href="#">Settings section</a>, <a href="#">Sequences section</a>, <a href="#">Functions section</a> and <a href="#">Other section</a> bars.</li> </ul> <p>The SeqNinja panels may be moved (<a href="#">Hide, show and change the sizes of Items in the SeqNinja interface</a>).</p>

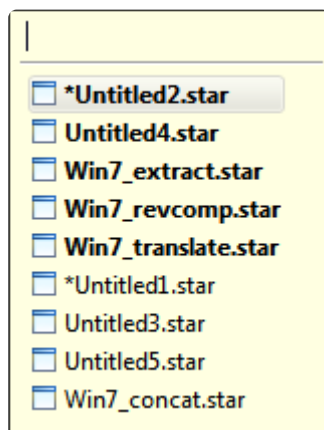
# The SeqNinja project

In SeqNinja, multiple projects are displayed in separate tabs, each assigned a unique name (Untitled[#].star).

- Left-click on any project tab to make that project active.
- Press **Ctrl+**; (Win) or **Cmd+**; (Mac) to open a “project chooser.” You may also open the project chooser by pressing the > > icon that appears to the right of the project tabs when a large number of projects are open.



The project chooser appears as a yellow popup with a line for typing in text at the top, and a list of open projects below. Unsaved projects that contain steps are marked with an asterisk (\*).

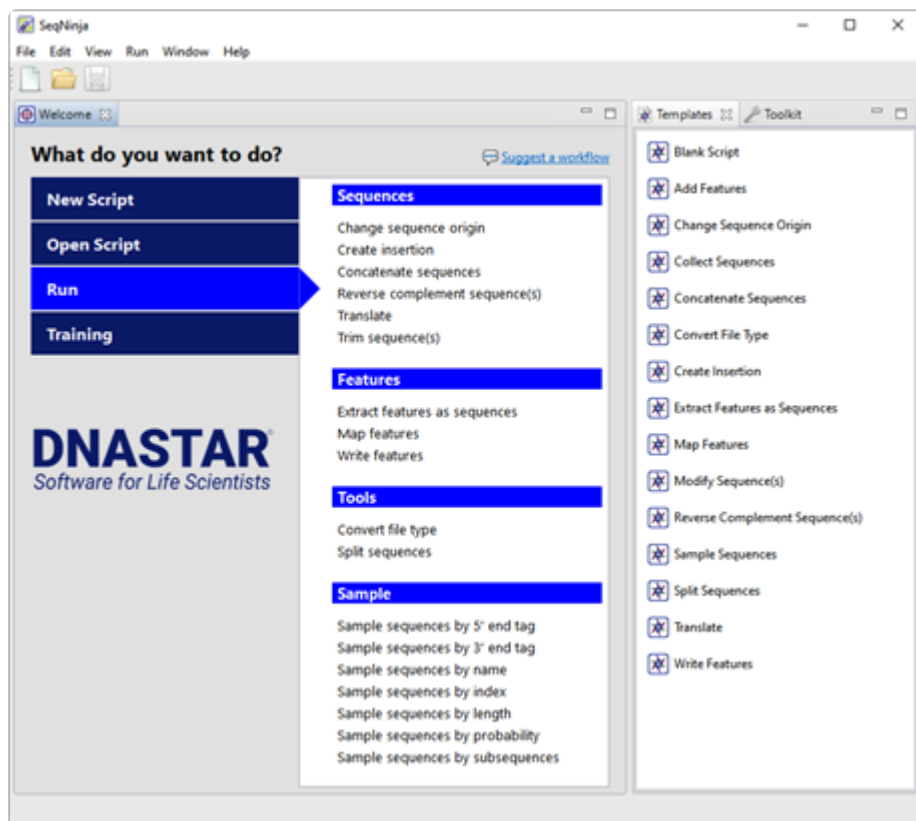


Use your mouse to click on the project you want to make active. If you do not have a mouse, make a selection using the keyboard arrow keys, then press **Enter**.

To instead search for a project using a filter, type text into the text field and press **Enter**. The text field supports the “?” and “\*” wildcards.

# Welcome screen

The Welcome “project” opens when you launch SeqNinja and is a central location for creating, opening, and running projects, and for accessing the User Guide. As you add projects, the Welcome “project” will remain as the leftmost tab unless you manually move or close it. We recommend leaving the Welcome tab in place, since it has many time-saving shortcuts to useful commands.



Four navy-blue tabs on the left reveal different sets of options, each of which is described below:

Tab on left	Option on right	Shortcut to this procedure:
New Script	New blank script	File > Script. (See <a href="#">Work with Sequences</a> ).
Template gallery	Shortcut to scripts in the <a href="#">Templates panel</a> .	
Open Script	Open	File > Open (See <a href="#">Work with Sequences</a> ).
	Recent documents	File > Recent Documents.
Run	Search for sequences	Shortcut to scripts in the <a href="#">Templates panel</a> .
Training	User guide	Opens this User Guide from the DNASTAR website.
	Tutorials	Opens the SeqNinja Tutorials from the DNASTAR

		website.
--	--	----------

# Script views

---

The script that SeqNinja will be running is created in either (or both) of the two script views:

- The [Visual view](#) is intended for beginning and intermediate SeqNinja users. It provides [Templates](#) for popular steps and a [Toolkit panel](#) for adding steps and operations to enhance the templates, or to create a new script from scratch.
- The [Text view](#) is intended for advanced users or those working with extremely long scripts and allows you to simply type the SeqNinja commands.

To begin a new script – See [Work with Sequences](#).


To navigate within the [Script views](#) – If you prefer navigating with a mouse, click anywhere on the script to move the mouse to that position. If you prefer keyboard navigation, use the **Tab** key or any of the arrow keys on your computer keyboard to navigate from one part of the script to another.

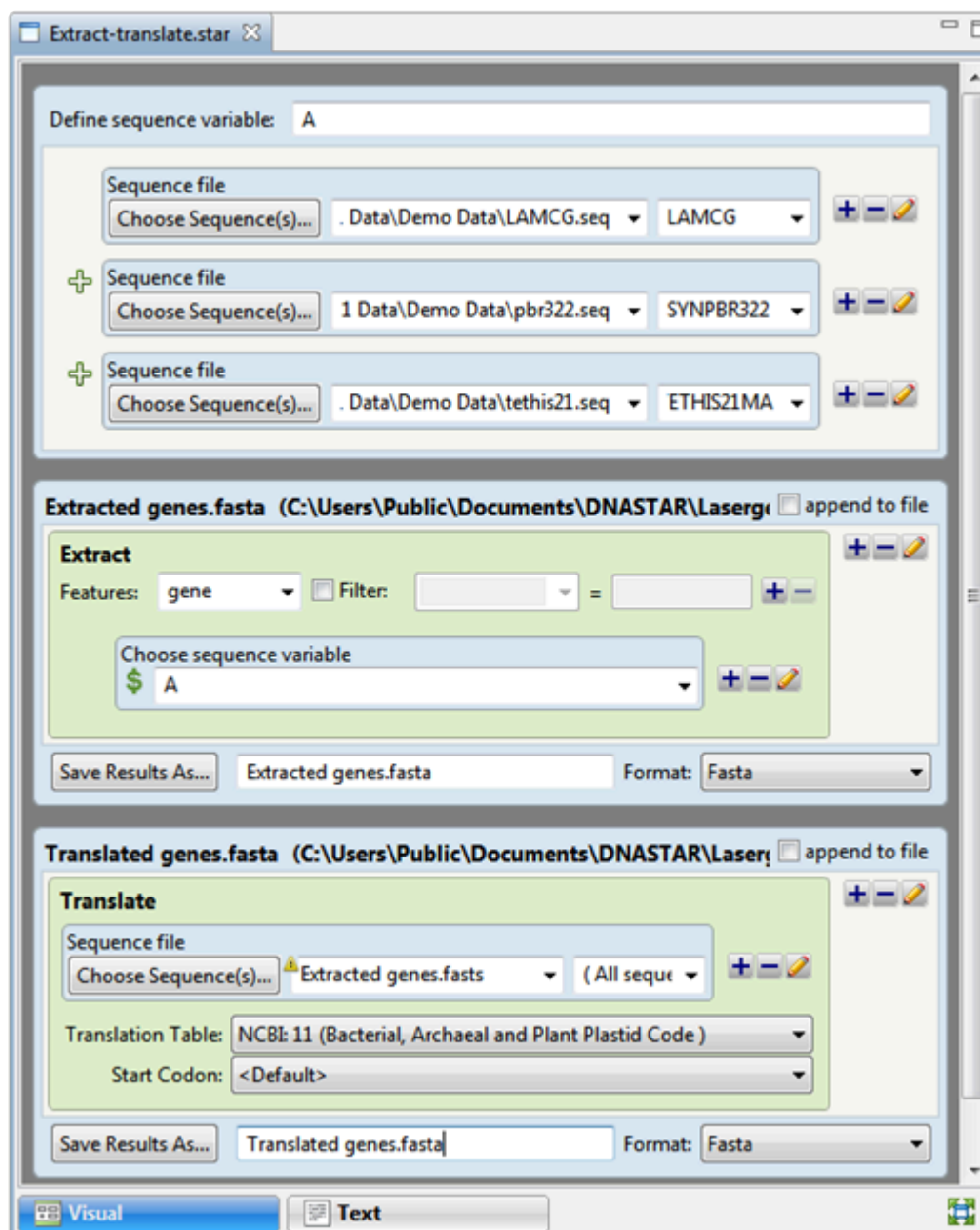
To delete text or numbers – Use the **Delete** key on your keyboard. If you make a selection from a drop-down menu and then change your mind and want to leave the drop-down menu blank, just click on the selection and then click the **Delete** key on your keyboard.

To learn about the SeqNinja toolbar tools – See [Toolbar](#).

# Visual view

The Visual view provides assistance and visualization during completion of a SeqNinja script. This view contains the same information as the [Text view](#), but in a graphical presentation. The Visual view is the default view.


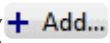
To open the Visual View, click on the **Visual** tab (  **Visual** ) or use the **View > Visual View** command. The following image shows the Visual view after adding sample [Templates panel](#) and [Toolkit panel](#) items and filling in the fields.



Entering data for an operation:

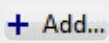

- To add a sequence, use the **Choose Sequence(s)** button. You may also type in the filename in the text box to its right (see [Add and modify a sequence](#)) or copy and paste in a filename from your clipboard. The sequence you enter will apply only within that sequence expression and not to subsequent operations.
- To use drop-down menus, click the arrow on the right of each menu, then select an item from the menu. Many drop-down menus allow you to type in text or numbers if the selection you want is not offered.
- To use text boxes, type in text, such as a filename or a feature type. For information on right-click shortcuts, see [Right-click functionality in the Visual view](#).

#### Appending a new operation to the Visual View (three methods):

- To add an operation to the bottom of the current script – Use any of these three sub-methods:
  - Click the Add icon (  ) to the right of the last operation in the Visual view.
  - Click the Add button (  ) below the last operation in the Visual view.
  - Click an item in the [Toolkit panel](#).
- To add an operation anywhere in the script, [drag & drop](#) an item from the [Templates panel](#) or [Toolkit panel](#) anywhere on the [Visual view](#).
- To add [Toolkit panel](#) items using right-click shortcuts, see [Right-click functionality in the Visual view](#).

#### Using the Visual view tools:



The following tools are available at the bottom right of the Visual view.

Tool	Function	Description
	Add	Clicking this button offers the following choices: <a href="#">Write to Results File</a> , <a href="#">Define Sequence Variable</a> , <a href="#">Add Comment</a> . For more information, see the Sequences section help topics <a href="#">Write to Results File</a> and <a href="#">Define Sequence Variable</a> ; and the Other section help topic <a href="#">Add Comment</a> .
	Run	Tells SeqNinja to run the script.



# Components of the visual view

This table shows definitions for components of the [Visual view](#).

SeqNinja Help Terminology		Scripting Language Terminology	Generated Via:
Top-Level Procedures	Lower-Level Procedures		
<p><b>Script</b> – A list, composed of one or more <b>steps</b>, <b>operations</b> or <b>comments</b> that can be executed using the <b>Run</b> button ().</p> <p>Steps are executed in the same order in which they appear in the script.</p> <p>A <a href="#">template</a> generates an “empty” script for a common task, and requires that values be filled in.</p>	<p><b>Step</b> – One or more <b>operations</b> appearing in a bounded box. Steps can be stacked to create a script.</p>	<b>Sequence Expression</b>	<a href="#">Add Sequence File</a> , <a href="#">Add Sequence Text</a> , and <a href="#">Add Sequence Variable</a> (all in the <a href="#">Sequences section</a> ) are sequence expressions. <a href="#">Add Sub-Range (Sequences section)</a> is a partial sequence expression.
	<p><b>Operation</b> – A single discrete command, such as “label this sequence as circular,” or “save results to this location.”</p>	<b>Function</b> – Denoted by a light green frame.	<a href="#">Functions section</a> items.
	<p>All <a href="#">Toolkit panel</a> items add either <b>steps</b> or <b>operations</b> to the script.</p>	<b>Setting</b> – Designates a location and/or option for subsequent use in the script. Each setting contains an equal sign (  .	<a href="#">Settings section</a> items  <a href="#">Define Integer Variable (Other section)</a> .
		<b>Action</b>	<a href="#">Print to Console (Other section)</a> .
		<b>Statement</b> – Denoted by a light blue frame.	<a href="#">Templates</a>  <a href="#">Write to Results File</a> , <a href="#">Define Sequence Variable</a> (both in the <a href="#">Sequences section</a> ).
	<p><b>Comment</b> – Editable text on a pale yellow background. Comments have no effect on the</p>		<a href="#">Add Comment (Other section)</a> .

	execution of a script.		
--	------------------------	--	--

# Right-click functionality in the Visual view

In the [Visual view](#), you can add or insert [Toolkit panel](#) items by right-clicking on certain operations or steps. The right-click menu is context-sensitive and will update to show the list of operations that are valid for the current selection.

Right-click on a text box to access shortcuts to the following **Edit** menu options:

Command	Function
<b>Cut</b>	To remove a portion of text and place it on the clipboard.
<b>Copy</b>	To copy a portion of text and place it on the clipboard.
<b>Paste</b>	To paste an item from the clipboard at the cursor insert point.
<b>Delete</b>	To delete a portion of text without placing it on the clipboard.
<b>Select All</b>	To select all text.

Right-click on any blue framed step to access the following commands:

Command	Function
<b>Write to Results File</b>	Shortcut to <a href="#">Write to Results File</a> , found in the <a href="#">Sequences section</a> .
<b>Define Sequence Variable</b>	Shortcut to <a href="#">Define Sequence Variable</a> , found in the <a href="#">Sequences section</a> .
<b>Add Comment</b>	Shortcut to <a href="#">Add Comment</a> , found in the <a href="#">Other section</a> .
<b>Assign to Variable / Write to File</b>	Shortcut to <a href="#">Add Sequence Variable</a> , found in the <a href="#">Sequences section</a> . The former command name appears when you have right-clicked on an operation that assigns a sequence to a file. The latter name appears when you have right-clicked on an operation that assigns a sequence to a variable.
<b>Delete Current Selection</b>	Deletes the step.

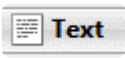
Right-click on any blue or green framed operation to access these commands:

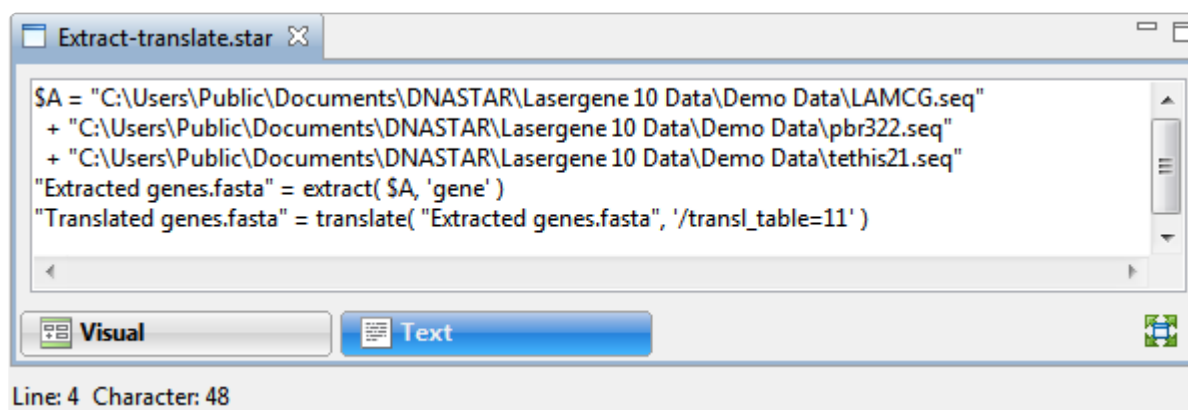
Command	Function
---------	----------

<b>Add Sub-Range</b>	Shortcut to <a href="#">Add Sub-Range</a> , found in the <a href="#">Sequences section</a> . This option is only available for blue framed operations.
<b>Add Sequence File</b>	Shortcut to <a href="#">Add Sequence File</a> , found in the <a href="#">Sequences section</a> .
<b>Add Sequence Text</b>	Shortcut to <a href="#">Add Sequence Text</a> , found in the <a href="#">Sequences section</a> .
<b>Add Sequence Variable</b>	Shortcut to <a href="#">Add Sequence Variable</a> , found in the <a href="#">Sequences section</a> .
<b>Apply Function &gt; FUNCTION</b>	The submenu provides shortcuts to each of the <a href="#">Functions section</a> items. Selecting a function from the <b>Apply Function</b> sub-menu applies that function to the original operation (i.e., wraps a green framed operation box around the original operation).
<b>Insert Function &gt; FUNCTION</b>	The submenu provides shortcuts to each of the <a href="#">Functions section</a> items. Selecting a function from the <b>Insert Function</b> sub-menu adds the function under the original operation (i.e., adds a green framed operation below the original operation).
<b>Remove FUNCTION</b>	Removes a function that was added using <b>Apply Function</b> or <b>Insert Function</b> .
<b>Delete Current Selection</b>	Deletes the operation.

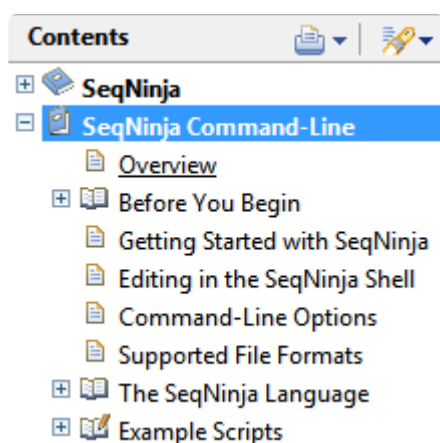
# Text view

This Text View contains the same information as the [Visual view](#), but in plain text. Advanced users may find it convenient to type or paste text into this view compared to using the Visual View. By simply typing or pasting text into the Text View (or copying and pasting text from the clipboard), you can enter top-level “statements” and lower-level “expressions” to formulate a script for SeqNinja.

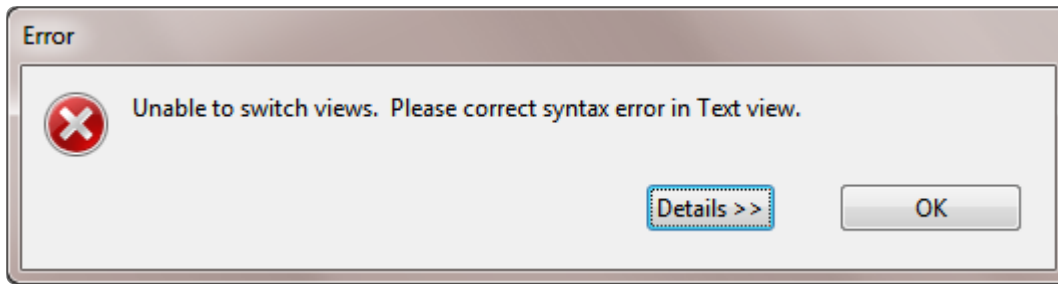
To open the Text View, click on the **Text** tab () or use the **View > Text View** command. The following image shows the Text view after entering a sample script. Note that the line and position of the cursor is displayed at the bottom left of the SeqNinja script, below the view tabs.



For detailed information about SeqNinja statements and expressions, see the [SeqNinja Command-Line Help](#).



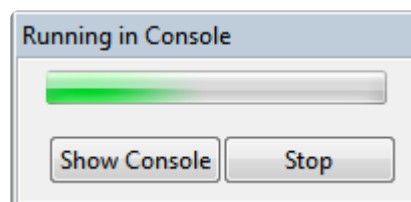
If you create a syntax error in the Text view and then attempt to switch to the Visual View, a warning will be displayed with details about the error. Click **OK**, then correct the error before switching to the Visual view.



None of the [Templates panel](#) or [Toolkit panel](#) options can be applied to the Text view, and the view does not support right-click functionality. However, if you have already begun a project in the Visual view, switching to this view will allow you to see what the script looks like in SeqNinja command language. From that point, you may continue by entering commands manually, or you can switch back to the Visual View to use the Templates and Toolkit.

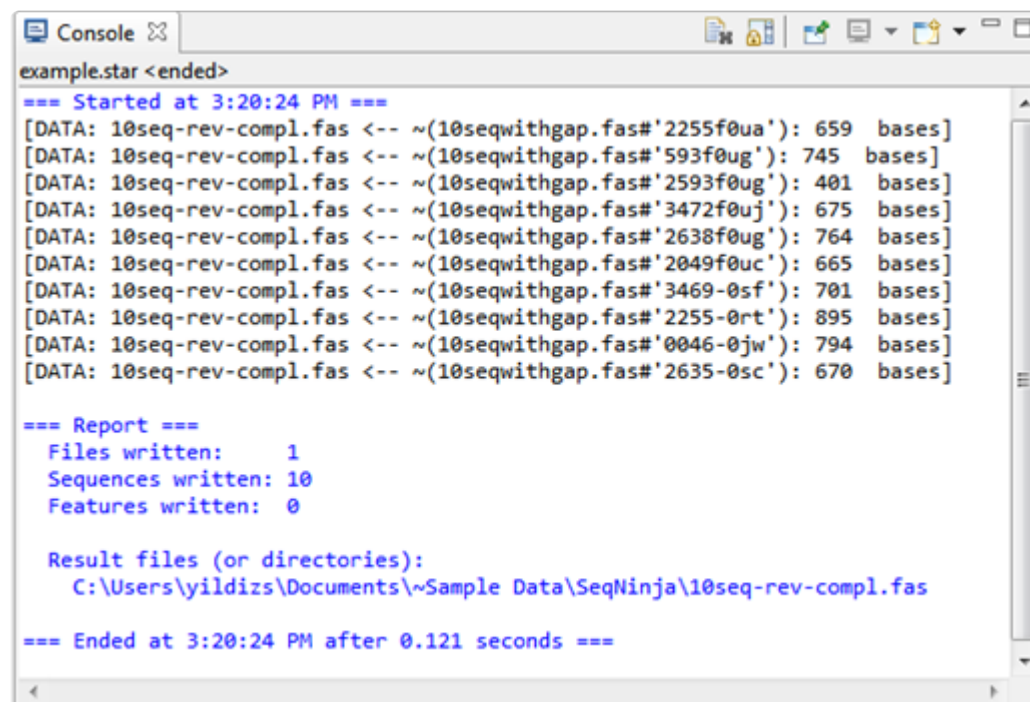
# Console panel

When you [run the script](#), the following dialog appears:



To view the Console, click the **Show Console** button. You can also open the Console using the **View > Console** menu command.







The Console panel displays one of a number of consoles, each representing a different run. Each Console displays the output of the run, accepts input, and shows the start and end times of the run.



Before and during a run, SeqNinja may ask you a question in the form of a pop-up dialog (see [Save, Close and Exit](#)). Your answer (e.g., “yes”) will be transcribed into the Console and the run will proceed or be terminated according to your response.

The following tools are available in the upper right of the Console panel:


Tool	Name	Description
	<b>Clear Console</b>	Clears all text from the active Console.

	<b>Scroll Lock</b>	Stops auto-scrolling caused by run output. When run output is long, may fail due to a limit on the amount of text in the console.
	<b>Pin Console</b>	Keeps the current Console active even if another script is selected.
	<b>Display Selected Console</b>	If multiple runs are in progress, clicking the arrow to the right of this icon opens a drop-down menu from which you can bring the desired Console to the front.
	<b>Open Console</b>	Opens a new Console. If you're running multiple scripts at the same time, you can open a new Console and pin it to a particular run using <b>Pin Console</b> (above).
	<b>Minimize</b>	Minimizes the Console.
	<b>Maximize</b>	Maximizes the Console.



# Toolbar

The tools located above the script tabs can be displayed or hidden using the commands **View > Show Toolbar** and **View > Hide Toolbar**.

Icon	Name	Description
	<b>New</b>	Shortcut for the <b>File &gt; New Script</b> menu command.
	<b>Open</b>	Shortcut for the <b>File &gt; Open</b> menu command.
	<b>Save</b>	Shortcut for the <b>File &gt; Save</b> menu command.
	<b>Run</b>	Tells SeqNinja to <a href="#">run the script</a> .
	<b>Stop</b>	Tells SeqNinja to <a href="#">stop running a script</a> .
	<b>Toggle the display of script labels</b>	Toggles the display of text labels in the script. More advanced users may wish to turn these off to create a more compact visual view. You can set the <a href="#">global default</a> for showing / hiding these labels using the command <b>Edit &gt; Preferences</b> (Win) or <b>SeqNinja &gt; Preferences</b> (Mac).

To learn about icons in the docking area, see [Hide, show and change the sizes of items in the SeqNinja interface](#).

# Templates panel

---

The Templates panel helps you quickly start a new SeqNinja script. You can also add a template to the [Visual view](#) of an existing script.

To open the Templates panel:

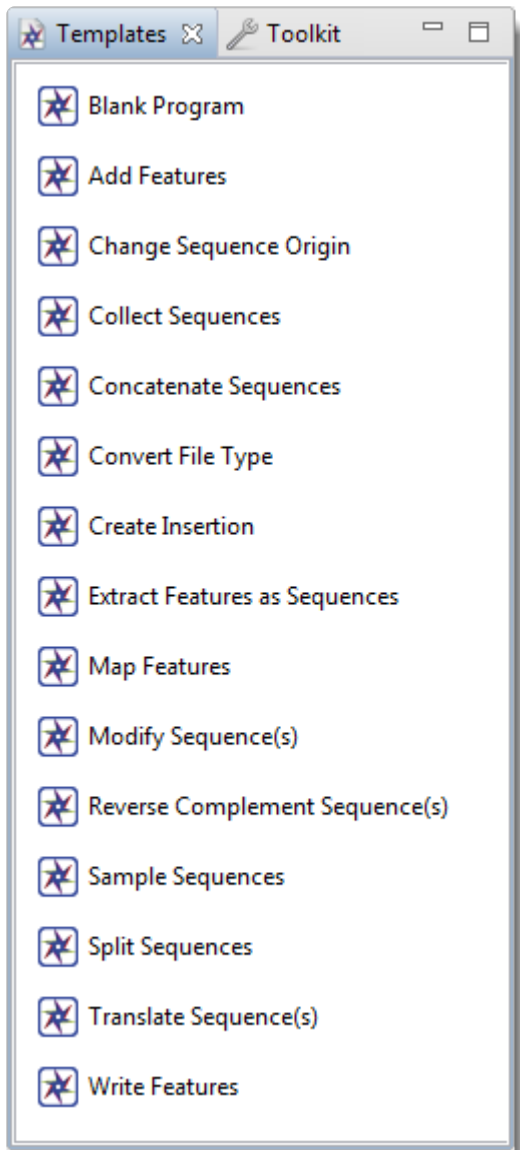
Use any of the following methods:

- Click on the **Templates** tab to bring the panel to the front.
- Choose **View > Templates**.
- Press **Ctrl+Alt+T** (Win) or **Cmd+Option+T** (Mac).

For information on opening, closing, minimizing, or maximizing the panel or its expand bars, see [Hide, show and change the sizes of items in the SeqNinja interface](#).

To open a new SeqNinja script with a template loaded

Click on the template in the SeqNinja Template Gallery or the Templates panel.



To insert a template anywhere in an existing SeqNinja script:

Drag & drop it from the Templates panel to the [Visual view](#).

# Blank Script

---

The Blank Script template, located in the [Templates panel](#), opens a new, blank script in the [Visual view](#). You can then add [Templates panel](#) or [Toolkit panel](#) items to create a script.

# Add Features

The Add Features template, located in the [Templates panel](#), lets you add specified features to the input sequences and saves the result. Alternatively, you can use this template to filter a variant call format file (.vcf) and save the results as a new VCF file.

Inputs consist of a sequence file (typically, one lacking features) and either a feature file (e.g., a *.starff* file) or a second sequence file that contains features. Note that if a second sequence is used, it can contain substitutions, compared to the sequence file, but may not have insertions or deletions.

The following examples illustrate some scenarios where this template may be useful:

- You want to add annotations to a sequence. With this template, you can define the annotations in a table of features (*.starff* feature file), add them to a sequence, and store the result. While annotation can be done manually in [SeqBuilder Pro](#), SeqNinja's Add Features template is useful for generating many features to a table format, then writing them to a sequence file format.
- You have two copies of the same biological sequence. One copy is a FASTA (*.fasta*) sequence with some capital letters that you wish to retain. The other copy is an annotated GenBank (*.gbk*) sequence. With Map Features, you can add the specified annotations from the GenBank file to the FASTA file, while still maintaining the capital letters in the latter.
- You have a large VCF file that contains extraneous data; for instance, data for all chromosomes when you are only interested in one particular chromosome. You can filter this large VCF file and save only the portion of it that corresponds to your area(s) of interest.

Adds features to one or more sequences.

Optionally, you can filter the features.

If the feature file identifies the sequences differently, provide comma-separated identifiers of the corresponding sequences in the feature file, in the order of the specified sequences.

☐ append to file

**Annotate**

Sequence file  
 Choose Sequence(s)... (All sequences) + -

With  
 All features

From  
 Feature file

Feature file: Browse...

☐ Sequence IDs in feature file:

Save Results As... Format:

- In the “Sequence file” area, use the **Choose Sequence(s)** button to specify the input sequence file that you wish to annotate (see [Add and modify a sequence](#)).
- Use the “With” drop-down menu to select which features you want to add to that sequence: **All features**, **Features matching** or **Features except**. If you choose anything other than **All features**, the step expands to contain a new row.

In the new row, select the feature type of interest (**gene**, **CDS**, **exon**, **intron**, **mRNA**, **tRNA**, **promoter** or **misc\_binding**) from the drop-down menu on the left, or type its name in the menu box. If you want to extract more than one feature type to the same output file, use the plus (+) button to add additional feature type lines. If you want to further limit the search to features matching (or not matching) particular qualifiers, check the **Filter** box, then choose a qualifier (**gene**, **/product**, **/locus\_tag**, **/note** or **/db\_xref**) from the drop-down menu to its right. In the right-most textbox, enter the text that the qualifier must match or not match (e.g., **/gene = thrL**) in order for the feature to be added to the sequence file. You may use wildcards in this box if you wish (e.g., **/gene = thr\***).

- Use the “From” drop-down menu to choose the type of file that will supply the annotations: **Feature file** or **Same sequence(s) with different features**.
- In the “Feature file” area, use the **Browse** button to navigate to the file.

✱ **Note:** If the feature file identifies sequences differently from the sequence file, you must provide comma-separated identifiers of the corresponding sequences in the feature file, in the order of the specified sequences.

- (optional) To allow a sequence ID in the feature file to differ from the ID in the sequence data, check the **Sequence IDs in feature file** box. Then use the text box on the right to specify comma-separated identifiers of corresponding sequences in the feature file. There is no need to reorder the IDs in the VCF file. Rather, the order of the identifiers must correspond to the order of sequences in the sequence data.

**Example 1:** You want to add VCF file features to a set of human chromosome sequences. However, the identifiers for chromosomes in the VCF file do not match the identifiers for the sequences. For example, a sequence might be identified in a GenBank file as NC\_000022, but in a VCF file as 22. Or perhaps one sequence file uses the names chr1, chr2, chrM; and the other uses 1, 2, MT.

**Example 2:** The sequences are ordered numerically {NC\_00001, NC\_00002,...} but the VCF variants are ordered alphabetically {1,10,11,...,2,20,...}. There is no need to re-order the variants in the VCF

file. Instead, enter the VCF sequence identifiers in the order of the sequences in the sequence file: { 1, 2, ..., 10, 11, ... }.

- In the **Save Results As** area, choose the name, location and format in which to save the output (see [Specify output format and location](#)). Choose the option “VCF” to write the output in variant call file format (.vcf).

# Change Sequence Origin

The Change Sequence Origin template, located in the [Templates panel](#), allows you to specify a new origin for the sequence. Later, when you [run the script](#), SeqNinja will reset the sequence origin using the range you entered and will output the results to the selected file.

- In the **Choose Sequence(s)** button row, choose the sequence for which you wish to reset the origin (see [Add and modify a sequence](#)).
- In the **Specify a range of the sequence** area, input a range, where the new origin is on the left, and the number on the right is one less (see [Set ends for a sequence](#)).
- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify output format and location](#)).

## Example:

Input:

GGAGTCTCCCATTGATTACA

Range: 10 > 9

Output:

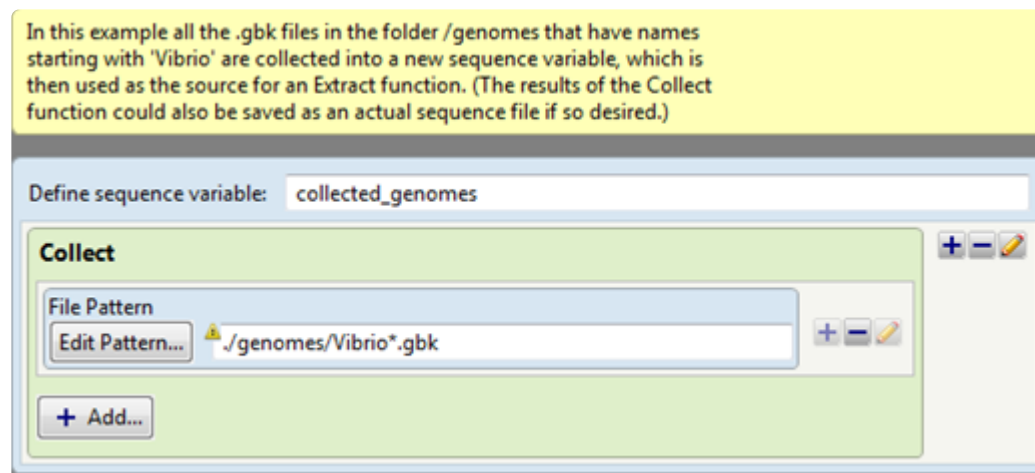
CATTGATTACAGGAGTCTCC



# Collect Sequences

The Collect Sequences template, located in the [Templates panel](#), allows you to collect all files in the same folder that share a particular naming pattern, and then either save them as a sequence file or convert them to a sequence variable to be referenced as a data source in later steps.

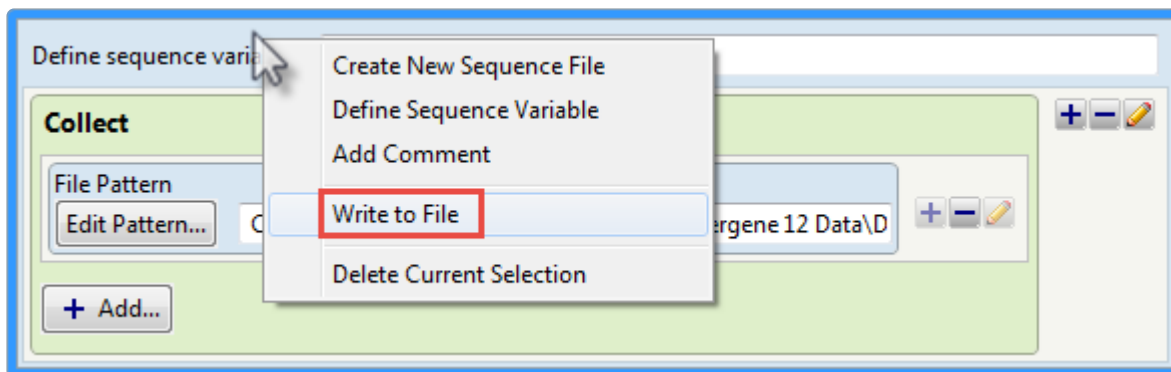
✿ **Note:** Collected sequences may later be split back into individual files using the [Split Sequences](#) tool.



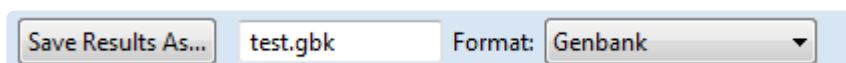
- Specify whether you wish to save the output as a sequence variable or a sequence file.
  - To save as a sequence variable: By default, the output is saved as a sequence variable named “collected\_genomes.” If you do not wish to use the default name, type a new name next to **Define sequence variable**. To use the output from this step as the input for a later step, you will simply reference the variable name you selected here.

Note: If you do not see **Define sequence variable** at the top of the step, bring it back by right-clicking on the blue frame around the step and choosing **Assign to Variable**.

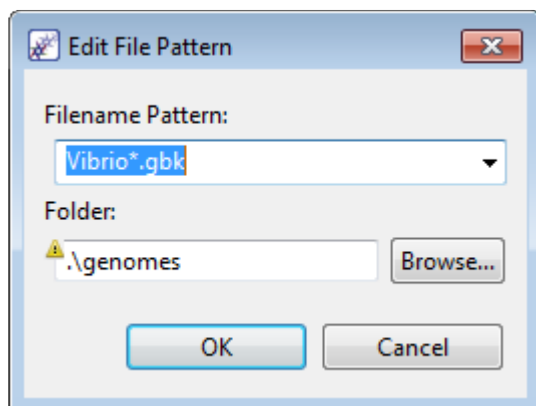
- To save as a sequence file: Right click on the blue frame around the step and choose **Write to File**.



The sequence variable input box at the top of the step disappears and is replaced with a “Save Results” section at the bottom of the step. If you wish to save to a non-default location, click **Save Results As**, navigate to the desired location for the output sequence, and then click **Save**. By default, the sequence is saved in GenBank format and is called test.gbk. To change the format, use the Format drop-down menu; to change the filename, enter a new filename in the center textbox.

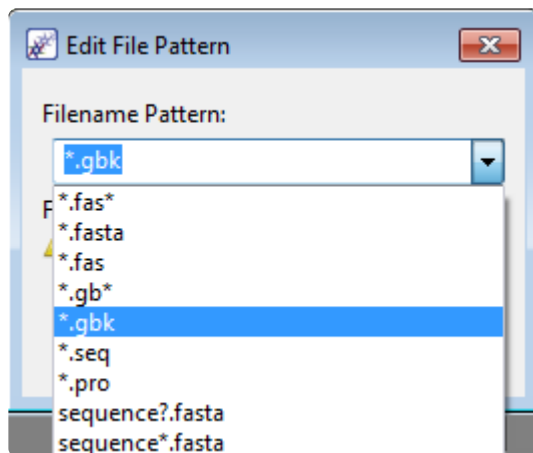


- To specify a pattern common to each of the sequence names, click the **Edit Pattern** button. This launches the Edit File Pattern dialog.



Within the Edit File Pattern dialog:

- Use the **Filename Pattern** dropdown menu to choose from a variety of file extensions and naming patterns.

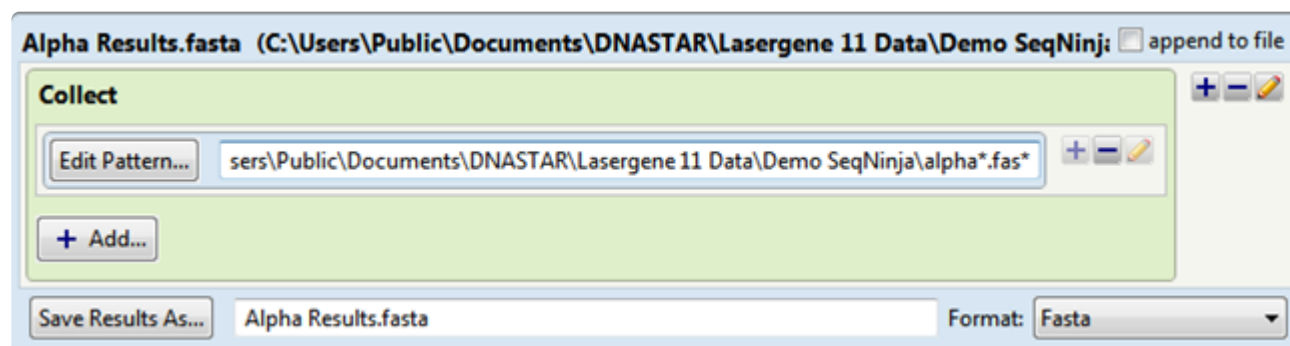


- In the **Filename Pattern** text box, type in the common portion of the filename for the files you wish to collect. Use an asterisk as a wildcard to signify which portion of the name varies between sequences.
- Click **OK** to return to the Collect Sequences template.
- The [Extract Features as Sequences](#) step is initially included as an “example step” after the Collect Sequences step. It can be used, removed (by right-clicking and choosing **Delete Current Selection**), or replaced with a different step, as desired.
- Click **Browse** and navigate to the folder containing the files that will be collected. After selecting the folder, press **OK** to close the Browse dialog and **OK** again to close the Edit File Pattern dialog.

#### Example 1:

If you wanted to collect two files named ysch2b1.seq and ysch2b2.seq, you would choose the .seq extension in Step 1. In Step 2, you would type **ysch2b** before the asterisk, yielding the following Filename Pattern text box entry: *ysch2b\*.seq*.

#### Example 2:



The goal is to collect all files in the specified directory that have names beginning with “alpha” and

extensions beginning with “fas.” The specified directory contains the files *alpha.fas*, *alphabet.fasta*, *alpha.fap*, and *beta.fas*.

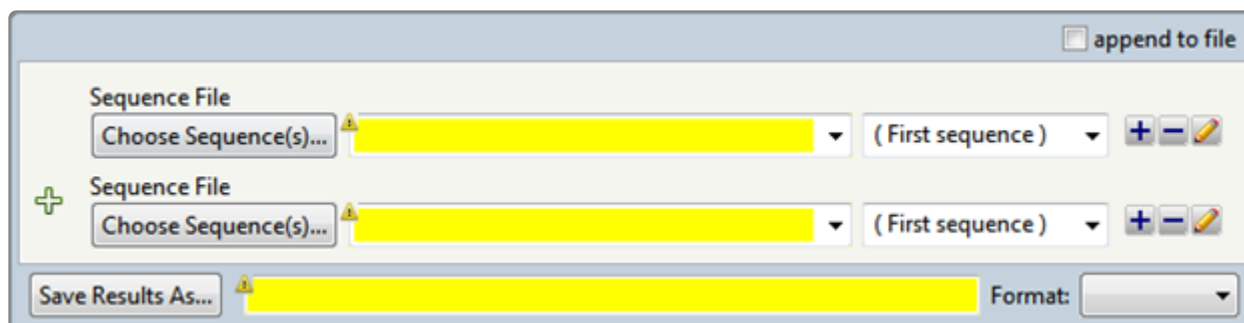
SeqNinja collects all the sequences from multi-sequence files *alpha.fas* and *alphabet.fasta* and saves them to the multi-sequence file *Alpha Results.fasta*.

# Concatenate Sequences

The Concatenate Sequences template, located in the [Templates panel](#), initially prompts you to enter two sequences and to choose the desired file output and location. You can also add additional items to be concatenated (e.g., additional sequences, sequence text strings) using the right-click menus or any of the [Sequences section](#) tools that start with the word “Add.”

Later, when you [run the script](#), SeqNinja will concatenate all combinations of the sequence(s) and/or text string(s) you entered and output the results to the selected file. The following rules apply:

- Any ranges added to this step are re-evaluated for each member of the set to which they are applied.
- The plus (+) operator concatenates, or links together, corresponding pairs in the left and right sequence sets.
- When one of the rows is a singleton (e.g. a literal such as “ATG”), the concatenation operation results in prepending or appending that single sequence to every sequence in the other row(s).



- In the **Choose Sequence(s)** button row, choose the sequences you wish to join (see [Add and modify a sequence](#)).
- Add additional sequences or modifiers, if desired, using the right-click menu or any of the [Sequences section](#) tools that start with the word “Add.”
- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify output format and location](#)).

## Examples:

### Simple concatenation example:

Input sequence 1: GGAGTCTCCC

Input sequence 2: ATTGATTACA

Output sequence: GGAGTCTCCCATTGATTACA

To add “ATG” to the beginning and “TAG” to the end of every sequence in a file:

Input sequence 1: ATG

Input sequence 2: mysequences.fasta

Input sequence 3: TAG

Output sequences: ATGcggtaaatacTAG, ATGgttaccaaTAG, etc.

To concatenate sub-ranges of sequences in a file:

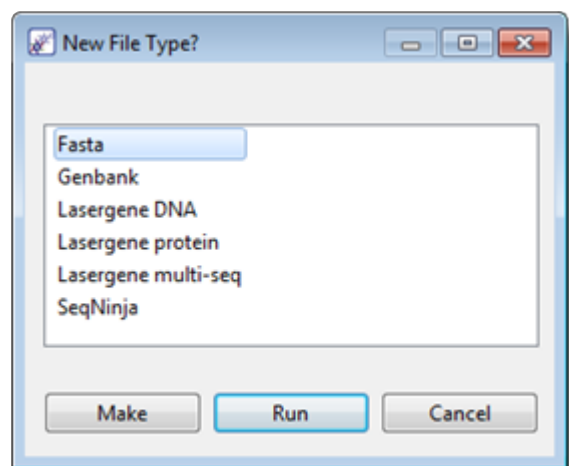
Input sequences: Sub-ranges 1-3 and 4-6 of the file two.fasta, which contains the following two sequences: gttacgtcgacgcgggcgtg and GTTACGTCGACGCGGGCGTG.

Output sequences: gttacg, GTTACG

# Convert File Type

The Convert File Type template, located in the [Templates panel](#), lets you convert a file from one format to another; for instance, from DNA to protein sequence.

When you select this template, you will immediately be prompted to open the source file. Once you have done so, the New File Type wizard will ask you to specify the output file type.



Press **Run** to fill in the template based on your selections and perform the conversion immediately or **Make** to fill in the template without running.

✿ **Note:** If you do not designate a file type different from the source sequence, SeqNinja will create a template specifying the same filename and location for both input and output. Clicking **Run** in this situation will generate an error message.

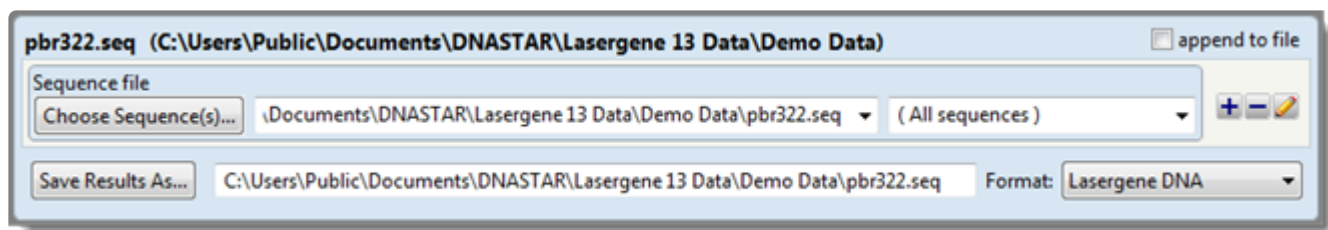
If you select **Make**, you can review the information and make any necessary changes before pressing **Run**:

- To change the source sequence, use the **Choose Sequence(s)** button (see [Add and modify a sequence](#)).
- To change the output format, make a selection from the **Format** drop-down menu.
- To specify the name and location in which to save the output (see [Specify output format and location](#)) press the **Save Results** button and select a location.

✿ **Note:** If you need to convert a file from .gbk to .fasta and back to .gbk while preserving features that would normally be lost during conversion to .fasta, prepend this command with the [Set Full File Conversion](#) step.

**Example:**

Input is as shown in the image.



After running the script, file *pBR322.gbk* is converted into the file *Sample.fasta*.



# Create Insertion

The Create Insertion template, located in the [Templates panel](#), prompts you to enter a sequence and to separately define the range of values that will go before and after the insertion, as well as to enter text for the insertion itself. Later, when you [run the script](#), SeqNinja will output a file consisting of the bracketing ranges and the insertion. This template is particularly useful if you need to migrate annotations between different versions of a genome while still preserving features.

Specify:  
 \* The segment before the insertion  
 \* The inserted sequence text  
 \* The segment after the insertion

For example, the template inserts "ACGT" between positions 3 and 4.

☐ append to file

Sequence file  
 Choose Sequence(s)... (All sequence: ...)

Specify a sub range of the sequence  
 ( 1 → 3 )

+ Type sequence  
 " ACGT "

+ Specify a sub range of the sequence  
 ( 4 → Right end )

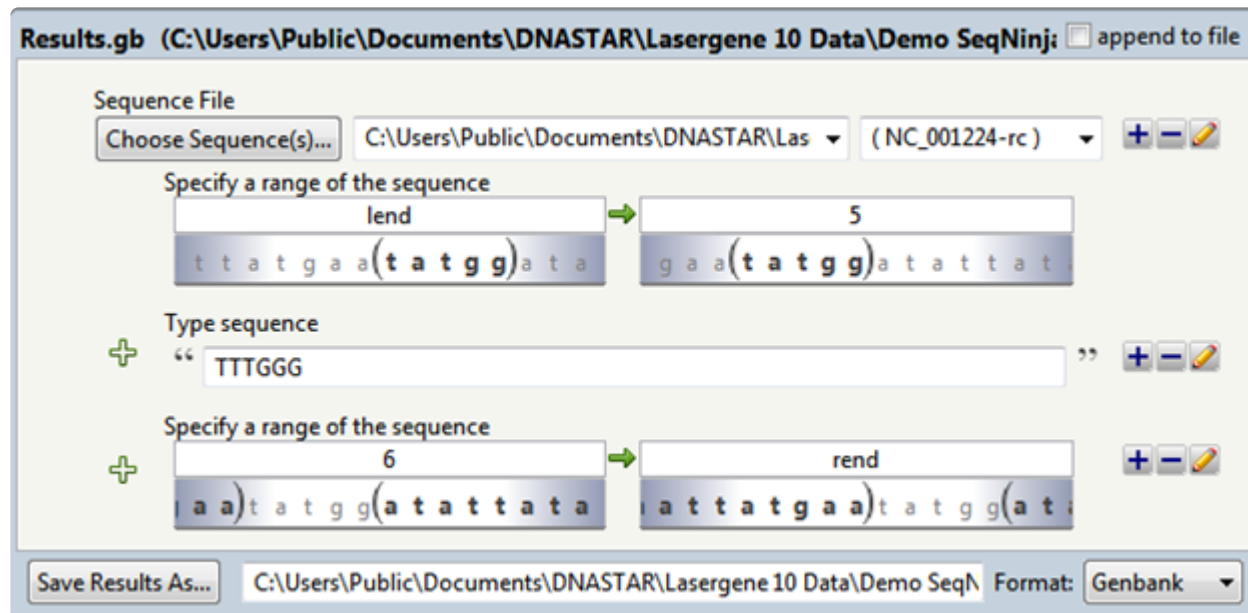
Save Results As... Format:

- In the **Choose Sequence(s)** button row, choose the sequence for which you wish to create an insertion (see [Add and modify a sequence](#)).
- In the upper **Specify a range of the sequence** area, input the first and last base number of the part of the sequence that will go before the insertion (see [Set sequence ends](#)).
- In the **Type sequence** text box, type the insertion. Only nucleotides and amino acids are allowed.
- In the lower **Specify a range of the sequence** area, input the first and last base number of the part of the sequence that will go after the insertion (see [Set sequence ends](#)).
- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify](#)

[output format and location](#)).

### Example:

Input sequence: GGAGTCTCCCATTGATTACA



Contents of the results file: GGAGTTTTGGGCTCCCATTGATTACA

# Extract Features as Sequences

The Extract Features as Sequences template, located in the [Templates panel](#), prompts you to choose the feature of interest, to enter an annotated sequence and a sequence range, and to choose the desired file output and location. Later, when you [run the script](#), SeqNinja will extract the features from the sequence and output the results to the selected file.

The screenshot shows a dialog box titled "Extract Features as Sequences". It has a tabbed interface with the "Extract" tab selected. The "Features:" dropdown is set to "CDS". There is a "Filter:" checkbox and a text box for a filter. The "Sequence File" section has a "Choose Sequence(s)..." button and a dropdown menu. The "Save Results As..." section has a text box and a "Format:" dropdown. There are also buttons for "+", "-", and a pencil icon.

- In the **Features** area, select the feature type of interest from the drop-down menu, or type its name in the menu box. If you want to extract more than one feature type to the same output file, use the plus (+) button to add additional feature type lines. If you want to further limit the search to features matching particular qualifiers, check the Filter box, then choose a qualifier from the drop-down menu to its right. In the right-most textbox, enter the text that the qualifier must match (e.g., */gene = thrL*). You may use wildcards in this box if you wish (e.g., */gene = thr\**).
- In the **Choose Sequence(s)** button row, choose the annotated sequence from which you wish to extract features (see [Add and modify a sequence](#)).
- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify output format and location](#)).

**Example:**

The screenshot shows the same dialog box as before, but with an example configuration. The "Features:" dropdown is set to "CDS". The "Filter:" checkbox is checked, and the filter text is "/gene = tet". The "Sequence File" section has a dropdown menu set to "gene 11 Data\Demo SeqNinja\pBR322.gb". The "Save Results As..." section has a text box containing "pBR322 Results.gb" and a "Format:" dropdown set to "Genbank".

The output is saved as pBR322 Results.gb, and the file contents are shown below:

```
> SYNPR322:tet  
ATGAAATCTAACAATGCGCTCA...
```

# Map Features

---

The Map Features template, located in the [Templates panel](#), lets you map chosen annotations from a source sequence to a target sequence using some of the same alignment algorithms used by MegAlign Pro. In cases where an input file contains more than one sequence (e.g., multiple chromosomes), only one sequence may be used at a time.

The Map Features template consists of a multi-screen wizard consisting of the [Feature](#), [Sequences](#), [Options](#) and [Output](#) screens. To navigate between the four wizard screens, either use the **Back** or **Next** buttons or click on the screen name in the left panel of the wizard.

The following are some scenarios where the Map Features template may be useful:

- You have two closely-related sequences, but one has missing annotations. With Map Features, you can map the desired annotations from the “complete” sequence to the “incomplete” one.
- You have a file containing features for the source sequence. With Map Features you can map the desired features to a second, closely-related, sequence.
- You want to map features from an old, annotated version of a sequence to a new, unannotated version.

SeqNinja can read from and write to a variety of source files, including FASTA (*.fas*, *.fap*, *.fasta*), GenBank (*.gbk*), DNASTAR Lasergene DNA and protein sequence files (*.seq*, *.pro*), DNASTAR Star Feature Files (*.starff*), and Variant Call Format files (*.vcf*). SeqNinja automatically recognizes the format of a source file by its extension.



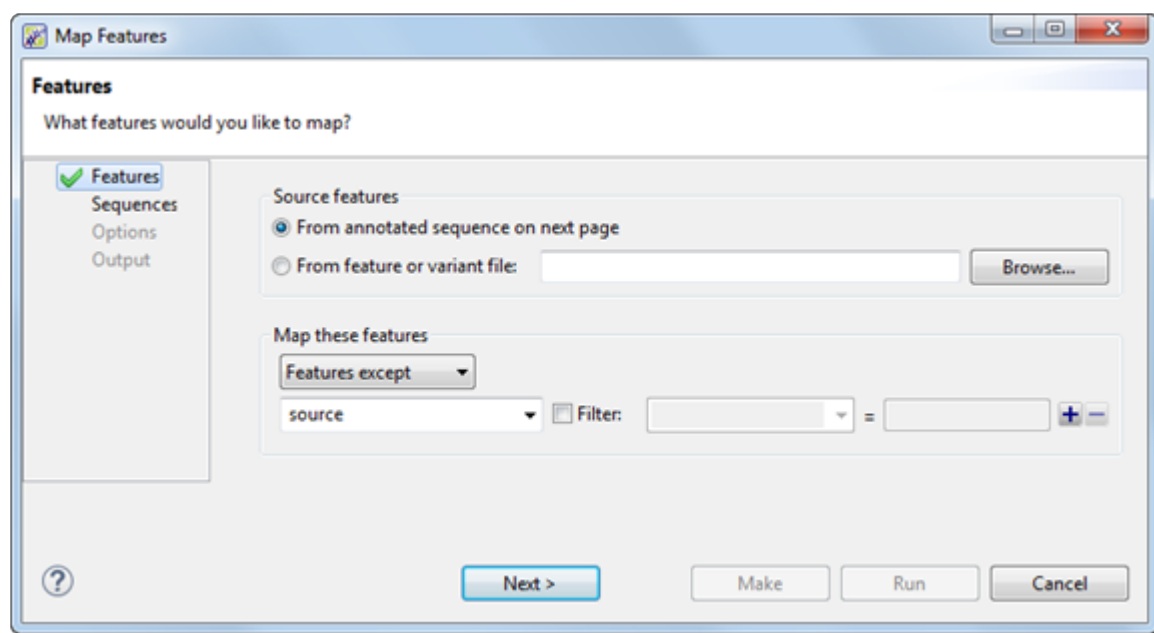
**Note:** If you wish to save to VCF format, you must use a VCF file as input.

A feature table may refer to chromosome sequences in any order and might contain a superset of sequences. SeqNinja automatically skips blocks of features for sequences that are not in the relevant sequence expression. For example, if you are only interested in chromosome 1, you can still use a VCF file containing variants for all the chromosomes.

# Features screen

The Features screen is the first wizard screen of the [Map Features](#) template and prompts you to choose the source of the features. The features may be from the source sequence (specified in the [Sequences screen](#)) or from a feature file (e.g., a VCF file). You can also specify which features you want to include or exclude when mapping to the target sequence.

This dialog is optional, and you may click **Next** to instead specify a source sequence on the next screen.



In the Source features section:

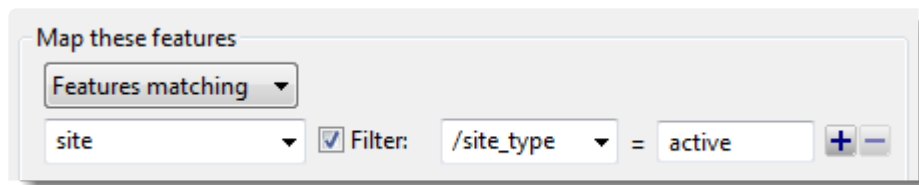
- If your source features are part of an annotated sequence, select **From annotated sequence on next page**. If they are in a feature or variant (VCF) file, select **From feature or variant file** and use the **Browse** button to navigate to the desired file.
- Use the upper drop-down menu to elect to map **All features**, **Features matching** those you will specify in the subsequent row(s) of the dialog, or **Features except** those specified in the subsequent row(s).

In the Map these features section:

Use the lower drop-down menu to select the feature type to match (or not match), e.g., **gene**, **CDS**, **exon**, **intron**, **mRNA**, **tRNA**, **promoter**, **misc\_binding**, etc. To specify more than one feature type, use the plus (+) button to add additional “Features” rows. If you want to further limit the search to features matching (or not matching) particular qualifiers, check the **Filter** box, then choose a qualifier (**gene**, **/product**, **/locus\_tag**, **/note** or **/db\_xref**) from the drop-down menu to its right. In the right-most textbox, enter the text

that the qualifier must match or *not* match (e.g., */gene = thrL*) in order for the feature to be removed in the output file. You may use wildcards in this box if you wish (e.g., */gene = thr\**).

The image below shows an example of how this section might appear after specifying the desired options:



Map these features

Features matching ▼

site ▼ ☒ Filter: /site\_type ▼ = active + -

Click **Next** to move to the [Sequences screen](#).

# Sequences screen

The Sequences screen is the second wizard screen of the [Map Features](#) template, and prompts you to specify the sequence type, source sequence and target sequence before proceeding to the next screen ([Options screen](#)).

- Use the **Sequence type** drop-down menu to choose the sequence type: **DNA**, **RNA** or **protein**. The source and target sequences must both be the same type.
- In the **Source sequence** area, use the **Browse** button to navigate to the file containing the source sequence. If the file contains multiple sequences, the first sequence is used by default. A different sequence can be selected using the drop-down menu. If the features come from a feature or variant file, you must enter it on the [Features screen](#). To return to that page, press the **Back** button, or click **Features** on the left of the wizard. You must still specify both a source and target sequence on the Sequences screen. If you specified a feature or variant file, the Sequences screen contains an extra row called **ID in feature file**. If the feature file uses the same name for the chromosome as the sequence file, this field can be left blank. Otherwise, enter the chromosome ID used in the feature file.
- In the **Target sequence** area, use the **Browse** button to navigate to the target sequence to which annotations will be added. If the file contains multiple sequences, the first sequence is used by default. A different sequence can be selected using the drop-down menu.

When all necessary information has been entered in both screens, click **Next** to move to the [Options screen](#).



# Options screen

The Options screen is the third wizard screen of the [Map Features](#) template, and lets you specify the alignment method and other mapping options.

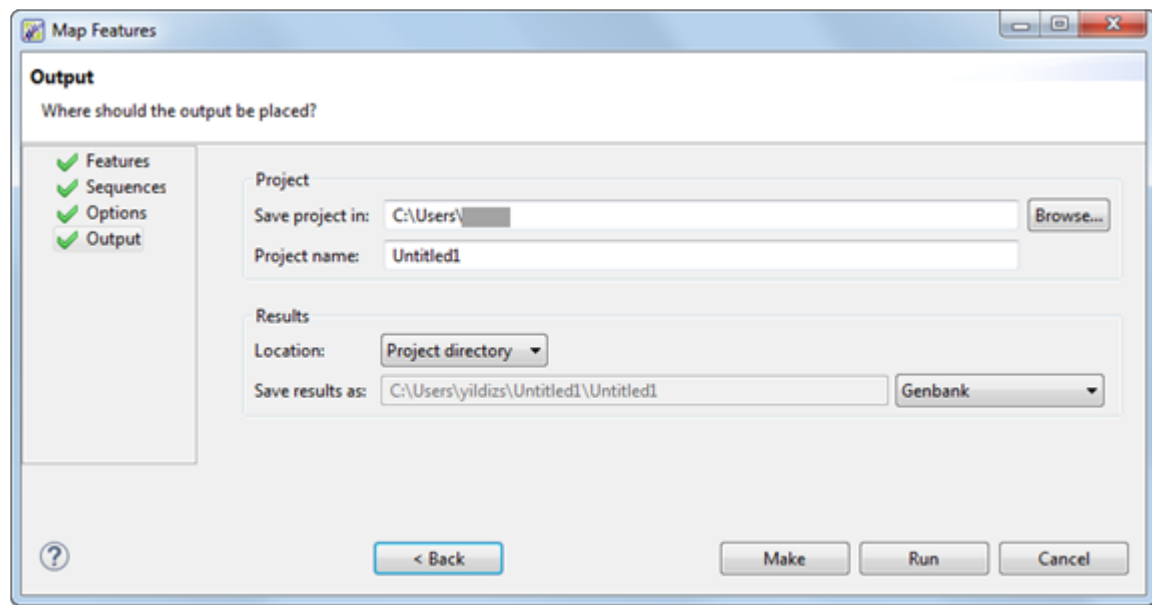
- In the **Align using** drop-down menu, choose between the multiple alignment methods **Clustal Omega**, **MAFFT**, **MUSCLE** and **Mauve**. For more information on each of these algorithms, see [this topic](#) of the online MegAlign Pro help.
- Under Mapping, use the text boxes or sliders to specify percent cutoffs for:
  - **Minimum feature coverage**, where:  $\text{feature coverage} = (\text{portion of feature contained within the aligned interval}) / (\text{feature length}) \times 100\%$
  - **Minimum sequence identity**, where:  $\text{sequence identity} = (\text{number of identical aligned residues}) / (\text{length of feature in gapped alignment space}) \times 100\%$ . In this calculation, non-common gaps are counted in both source and target sequences. Feature coverage is less sensitive to insertions and substitutions than is sequence identity. The default for both Mapping items is 80%.

- The genetic code is used to create translation annotations for coding sequences. It is also used to characterize variation reported in the report file (see next bullet point). By default, SeqNinja translates using the genetic code identified in the source data (e.g., the GenBank /transl\_table qualifier), if available; otherwise, it uses the standard genetic code (NCBI:1). To specify a different code, check **Override genetic code** and make a selection from the corresponding drop-down menu.
- After running the Map Features script, the output is sent to the [Map Features Report](#) file. To add a column to that report containing a SeqNinja expression describing the bases in the mapped feature in terms of the source and target sequences, check the box next to **Include specific sequence changes**.
- To create an Unmapped Interval Report listing regions that may need further annotation, check the box next to **Generate report of intervals without mapped features**. Then use the text box **Minimum length of interval to report** to specify a minimum length for a stretch of sequence where features were not mapped.

Click **Next** to proceed to the [Output screen](#).

# Output screen

The Output screen is the fourth wizard screen of the [Map Features](#) template, and lets you specify where to store the output of the feature mapping. This output includes the feature mapping script, reports and logs, and the result sequence.




- In the **Save project in** box, specify the directory that will contain the project directory. To use a different directory than the one listed, use the **Browse** button.
- In the **Project name** box, type in a name for the project. The project directory will be created in the specified folder, with the specified name. The project directory cannot be an existing directory. Therefore, if you decide to repeat the mapping using different options, you will need to change the project directory before re-running.
- After running the script, the result sequence will be created in the project directory, by default. If you would like to store it elsewhere, select **Custom** from the **Location** drop-down menu, and specify the desired result file in the **Save results as** row. You may use the **Browse** button to navigate to the desired location.
- Use the drop-down menu to the right of **Save results as** to choose a file format for the result output.

When you have made your selections in this screen:

- Click **Run** to map features and generate the result sequence. SeqNinja will extract the features from the input sequence and add the filtered annotations to the target sequence. While features are being mapped, you can press the Stop button to cancel mapping. The dialog will stay open, so that you can make changes and start mapping again. When work is complete, you can press the Close button to

exit the dialog. To learn about the files created after running this script, see [Map Features output files](#).

- Click the **Make** button to save these options as a reusable script that can be run later to generate the result sequence, with or without first making additional changes to the script (e.g., to use other sequences as input, etc.). **Example:** After mapping features, you notice that too many features did not satisfy the coverage and identity constraints specified on the [Options screen](#). You can lower the thresholds on the Options screen, specify a new project directory, and re-run.

 **Note:** Sequences are aligned, and feature mapping is determined regardless of whether you choose **Run** or **Make**.

- Click **Cancel** to leave the dialog without doing any of these.

# Map Features output files


After running the [Map Features](#) step, the output consists of approximately seven files of several file types.

File Name	Notes
<b><i>PROJECT.EXT</i></b>	Results file containing the mapped features. Can optionally be generated outside the project directory. Default is Genbank (.gb).
<b><i>PROJECT_map_features.star</i></b>	SeqNinja script that adds the annotations in the .starff table to the target sequence.
<b><i>PROJECT_map_features.starff</i></b>	Table of mapped features which are added to the target sequence when the Run button is pressed in the Map Features dialog, or when the SeqNinja script _PROJECT_map_features.star is loaded and run.
<b><i>PROJECT_map_features_console.txt</i></b>	Copy of the text that appeared for this mapping in the <a href="#">Console panel</a> in the SeqNinja user interface.
<b><i>PROJECT_map_features_report.tsv</i></b>	See the text and table below this one for additional information.
<b><i>PROJECT_map_features_unmapped.tsv</i></b>	Report of intervals in the target sequence to which no features were mapped. Generated only if requested in the <a href="#">Map Features Options screen</a> under <b>Unmapped Interval Report</b> .
<b><i>PROJECT_seqmover.star</i></b>	SeqNinja script that describes the target sequence in terms of segments common with the source, and segments unique to the target.

The report file is a tab-separated report file that can be opened in a spreadsheet (recommended) or a text editor. This report file contains a row for each feature that you directed SeqNinja to map. The “Variation” column summarizes the most significant variation of the mapped feature against the source feature:

Category	Label in report	Comments
<b>Not mapped</b>	Not_mapped	Can occur when the feature doesn’t map at all, or when the feature maps, but not well enough to pass the mapping thresholds.
<b>Indels</b>	Deleted_end_5prime Deleted_end_3prime Deleted_end Deleted_ends	[Non-protein sequences] First residue and/or last residue does not map or aligns with a target gap.
	Indel	Both insertions and deletions have occurred
	Insertion	An insertion has occurred

	Deletion	A deletion has occurred.
<b>Substitutions</b>	No_start	[CDS only] The initial codon in the mapped feature is not a start codon.
	Nonsense	[CDS only] A substitution has resulted in an interior stop codon.
	No_stop	[CDS only] The last codon in the mapped feature is not a stop codon.
	Non-synonymous	[CDS only] Non-synonymous substitution occurred.
	Synonymous	[CDS only] Synonymous substitution occurred.
	Substitution	[Non-CDS] A substitution has occurred.
<b>No change</b>	Identical	

 **Note:** If more than one item in the above table applies, the topmost item is reported.

# Feature mapping comparison: MegAlign Pro vs. SeqNinja

---

The ability to [map features](#) exists in both [MegAlign Pro](#) and SeqNinja. Here are some reasons you might want to map features in one application versus the other:

If you want to:	Use this application:	
	MegAlign Pro	SeqNinja
View the alignment and the mapped features	x	
Configure alignment parameters	x	
Map features in an end-to-end alignment. For instance, to perform a Mauve alignment of end-to-end chromosomes with some rearrangement.	x	
Map VCF features from a .vcf file.		x
Map features into a table, then edit the table before generating the resulting sequence or feature file		x

# Modify Sequence(s)

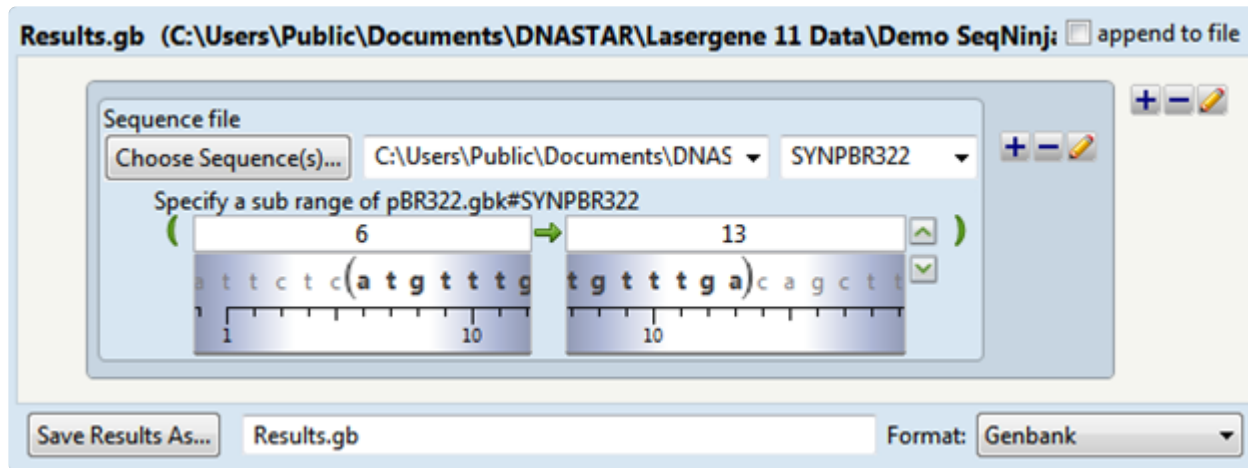
The Modify Sequences template, located in the [Templates panel](#), prompts you to enter a sequence and sequence range, and to choose the desired file output and location. Using the [Toolkit panel](#), you can optionally add more ranges and/or insert additional sequence text. Later, when you [run the script](#), SeqNinja will save the output to the selected file.

- In the **Choose Sequence(s)** button row, choose the sequence for which you wish to specify a sub-range (see [Add and modify a sequence](#)).
- In the **Specify a range of the sequence** area, input the desired range (see [Set ends for a sequence](#)).
- (optional) Use the Toolkit if you wish to add more ranges and/or insert additional sequence text.
- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify output format and location](#)).

## Example:

Sequence: TTCTCATGTTTGACAGCTTCGCGTTGGG





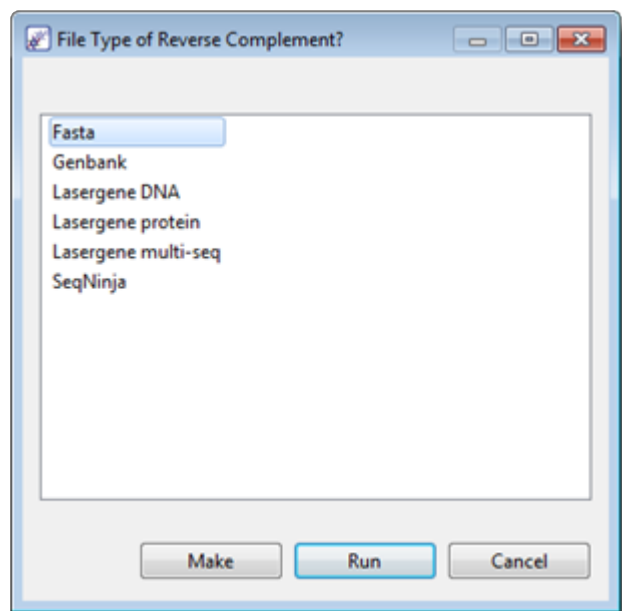
Results are saved as *Results.gb*, and the contents of the file are shown below:

ATGTTTGA

# Reverse Complement Sequence(s)

The Reverse Complement Sequence(s) template, located in the [Templates panel](#), lets you choose a source file and choose the desired file output and location. Later, when you [run the script](#), SeqNinja will create an output sequence that is the reverse complement of the source file.

When you select this template, you will immediately be prompted to select the source (input) file. Once you have done so, the File Type of Reverse Complement wizard will ask you to choose the desired output file type from a list.



Press **Run** to fill in the template based on your selections and run the script immediately or **Make** to fill in the template without running.

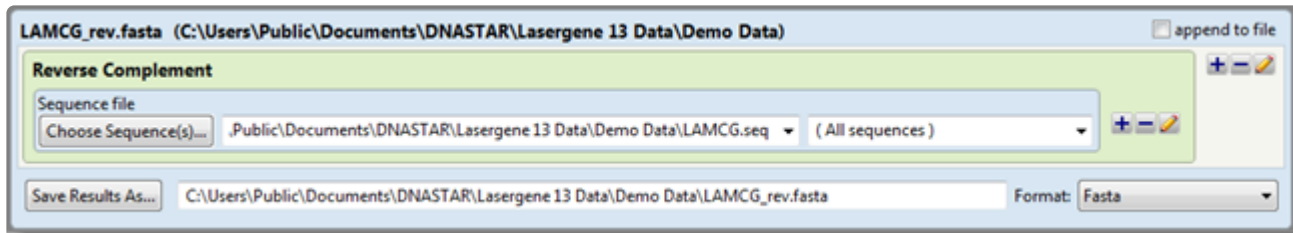
Choosing **Make** lets you review the information and make any necessary changes before pressing **Run**:

- To change the source sequence, use the **Choose Sequence(s)** button (see [Add and modify a sequence](#)).
- To change the output format, make a selection from the **Format** drop-down menu.
- To specify the name and location in which to save the output (see [Specify output format and location](#)) press the **Save Results** button and select a location.

✿ **Note:** If you need to convert a file from *.gbk* to *.fasta* and back to *.gbk* while preserving features that would normally be lost during conversion to *.fasta*, prepend this command with SeqNinja's [Set Full File Conversion](#) step.

After setting up the template as desired, click **Run** to run the script and create an output file. Or choose **Cancel** to exit from the template without running the script.

### Example:



*LAMCG.seq* begins with the string: GGGCGGCGACCTCGCGGGTTTTCGCTATTTATGAAA.

Output file *LAMCG\_rev.fasta* contains the reverse complemented string, which ends with: TTTTCATAAATAGCGAAAACCCGCGAGGTCGCCGCC.

# Sample Sequences

The Sample Sequences template, located in the [Templates panel](#), is used to make an output file that contains a filtered set of sequences from the source file. Source file sequences can be filtered according to one or more specified conditions, such as length, contents, and start/end sequence characters.

Initially, template options are pre-selected (or pre-filled) to show an example of how to filter for sequences at least 375 nt in length and containing the sequence “GATCT.” It is intended that you overwrite these selections to fit your own needs.

Sampling is achieved by applying one or more filtering steps on the input sequence set. Any combination of filters can be used, in any order, although each filter can only be used once in a given statement. See the help for a description of the different filters.

In this example, the filtering is for sequences that are at least 375 nt long, and that contain the sequence "GATCT". Note that the Contains filter will find sequences containing the specified sequence string on either strand.

☐ append to file

**Sample**

Filter: Minimum Length Value: 375 + -

Filter: Contains Value: GATCT + -

Sequence file

Choose Sequence(s)... (All sequence) + -

Save Results As... Format:


+ Add... Run

- One or more filter rows are needed in order to specify the sampling criteria. Two Filter rows have been provided as examples and can be edited or removed.
  - To delete a Filter row or add a new one, click on the plus or minus tools (+ -) on the right of each row.
  - To edit a Filter row, make selections from the **Filter** drop-down menus and filling in the corresponding **Value** boxes. The **Filter** drop-down menus offer the following options:

Use this filter:	To include:	Allowable values
<b>Minimum Length</b>	Only sequences the same or longer than the specified length.	Positive integer
<b>Contains</b>	Only sequences containing a specified sequence fragment.	DNA or protein

	For sequences of DNA or unknown type, matches can occur on either strand.	sequence fragment using 1-letter IUPAC codes.
<b>From Sequence Index</b>	All sequences beginning with the sequence of this name.	Sequence name
<b>To Sequence Index</b>	All sequences up to and including the sequence of this name.	Sequence name
<b>Sample Every</b>	Every 'nth' sequence in the source file, where 'n' is a positive integer.	Positive integer
<b>Sequence Name</b>	All sequences with this name.	Sequence name
<b>Probability to Include</b>	A random subset of sequences. Each member of the source set individually has the given probability of being included.	A single-quoted decimal value from 0.0-1.0 (e.g., '0.7')
<b>Maximum Length</b>	Only sequences the same or shorter than the specified length.	Positive integer
<b>Starts With</b>	Only sequences beginning with a specified sequence fragment. For sequences of DNA or unknown type, matches can occur at the beginning of either strand.	DNA or protein sequence fragment using 1-letter IUPAC codes.
<b>Ends With</b>	Only sequences ending with a specified sequence fragment. For sequences of DNA or unknown type, matches can occur at the end of either strand.	DNA or protein sequence fragment using 1-letter IUPAC codes.

- In the **Choose Sequence(s)** button row, enter the sequence you wish to sample (see [Add and modify a sequence](#)).
- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify output format and location](#)).
- Use the **Format** drop-down menu to select the file type for the output file.

 **Note:** To sample all sequences that do NOT start with, end with, or contain a specified subsequence, fill in the template as if you want to sample sequences that do start with, end with, or contain that subsequence. Then switch to the [Text view](#) and place a negative operator (!) before the argument.

**Example #1 input:**
**Example #1 output:**

A .fasta file containing the name and length of each output sequence (sequences #1, 3, 5, 7 and 9), followed by the sequence itself.

**Example #2 input:**
**Example #2 output:**

All sequences from contig01.fas that contain the sequence segment "TTGTT" have the bases "ATG" added to the beginnings of their sequences and "TAG" added to the ends of their sequences.

# Split Sequences

The Split Sequences template, located in the [Templates panel](#), is used to save a new, individual output file for each of the sequences in the source file.

The asterisk in the output pattern is required, as each sequence in the input file will be a separate output file, in the format specified, with the sequence name used as the file name. Prefixes and/or suffixes can also be specified to the pattern; in the example below, "new\_" will be added to the beginning of each sequence name to generate the file name.

To prevent large numbers of files from being inadvertently generated, a limit of 1000 sequence files is set as a default; this can be changed.

**Split Sequence Set**

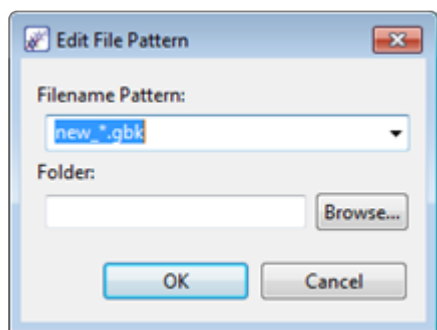
Sequence file  
  (All sequences)

Output Pattern:  new\_\*.gbk Limit Files To: 1000

- In the **Choose Sequence(s)** button row, enter the source sequence (see [Add and modify a sequence](#)).
- In the **Output Pattern** row, specify the pattern to use when naming the output files. The output file pattern must contain an asterisk (\*) to the left of the dot, but text may be used on either side of the asterisk. The file pattern can be defined in either of two ways:

## Method 1

1. Press the **Edit Pattern** button.



2. To change the file type from the default of Genbank, click the arrow to the right of "new\_\*.gbk"

to choose from a variety of file extensions.

3. In the **Filename Pattern** text box, type the desired text, making sure to retain an asterisk anywhere to the left of the dot.
4. Unless otherwise specified, output files will be saved in the [default directory](#). To select a different folder, use the **Browse** button.

## Method 2

Type in a new pattern and/or file extension in the Filename Pattern box. This option does not allow you to specify a non-default folder for the output files.

- The **Limit Files To** box contains a default of 1000 sequences in order to prevent a large number of files from being inadvertently generated. The limit can be changed, if desired, by typing a different number into the box



# Translate

The Translate template, located in the [Templates panel](#), prompts you to enter a sequence and a translation table, and to choose the desired file output and location. Later, when you [run the script](#), SeqNinja will translate the nucleic acid sequence into protein sequence using the selected translation table, then output the results to the selected file.

If the input sequence is annotated, each CDS feature will be translated separately, and any translation table and/or codon\_start annotations will be honored. If the input is unannotated, or contains no CDS features, the entire sequence will be translated. Note that translation only progresses for sequences with lengths that are multiples of three (i.e., codons). An “extra” base or two at the end will not be reflected in the output.

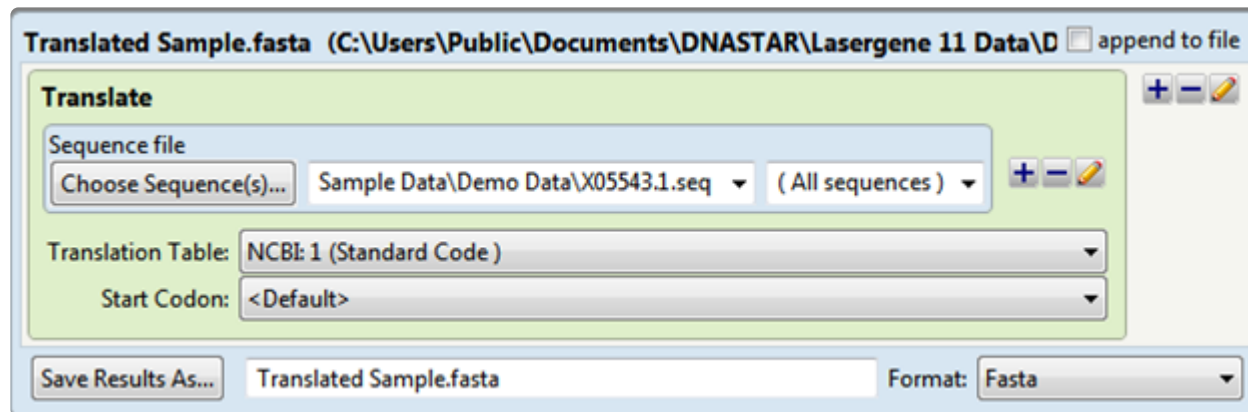
- In the **Choose Sequence(s)** button row, enter the sequence you wish to translate (see [Add and modify a sequence](#)).
- In the **Translation Table** area, choose a table from the drop-down menu. All NCBI and Lasergene tables are available and are denoted by number and/or description. By default, SeqNinja translates using the genetic code identified in the source data (e.g., the GenBank /transl\_table qualifier), if available; otherwise, it uses the standard genetic code (NCBI:1).
- Use the **Reading Frame** drop-down menu to choose the reading frame for the translation.
  - **<Default: 1 or as annotated>** directs SeqNinja to honor any existing /codon\_start annotations in the input file(s); or otherwise, to begin at the first base.
  - **Ignore '/codon\_start' annotations** directs SeqNinja to begin translating at the first base and to ignore any /codon\_start sequence annotations. Note that this selection is equivalent to the **<Default: 1 or as annotated>** selection (above) for any input sequence other than a .gbk or .seq file with /codon\_start qualifiers.
  - **1, 2 and 3** direct SeqNinja to begin the translation at the first, second or third base, respectively, ignoring any /codon\_start annotations. For an input sequence containing /codon\_start qualifiers, selecting the start codon implicitly is similar to choosing **Ignore '/codon\_start' annotations**,

but allows a specific frame to be set.

- In the **Save Results As** area, choose the name and location in which to save the output (see [Specify output format and location](#)).

### Example:

Input sequence: GGAGTCTCCCATTGATTACA



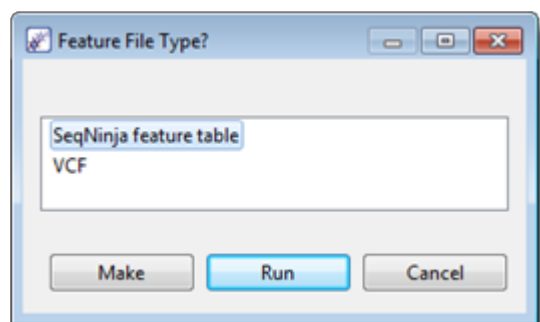
Results are saved as *Translated Sample.fasta*, and the contents of the file are shown below:

GVSH.L

# Write Features

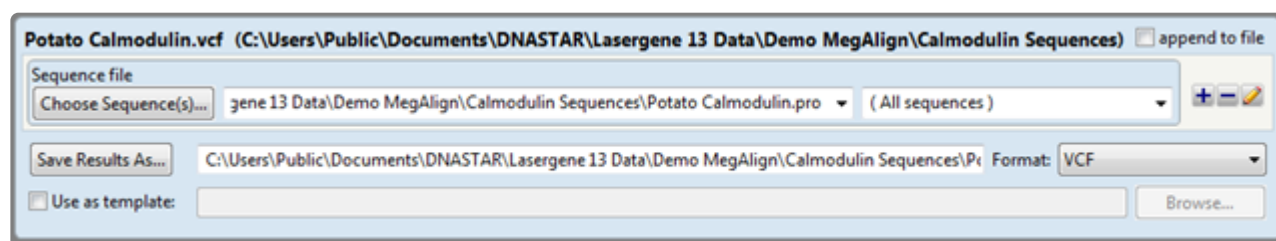
The Write Features template, located in the [Templates panel](#), is used to create a SeqNinja feature table or VCF file using the annotations from a specified sequence file. The output features files can be used with other SeqNinja scripts, such as [Map Features](#).

When you choose this template, you are immediately prompted to open a sequence from a file chooser dialog. After making a selection, the Feature File Type dialog appears. Make a choice from the list.



Press **Run** to fill in the template based on your selections and create the feature file immediately or **Make** to fill in the template without running.

Choose **Make** to review the information and make any necessary changes before pressing **Run**:



- To change the source sequence, use the **Choose Sequence(s)** button (see [Add and modify a sequence](#)). To select a particular sequence from the file, rather than the entire file (default), make a selection from the adjacent drop-down menu.
- To change the output format, make a selection from the **Format** drop-down menu. Most available options will save the output as a sequence file. The final two options—\*SeqNinja Feature Table\* and \*VCF\*—will instead save the output as a feature file.
- To specify the name and location in which to save the output (see [Specify output format and location](#)) press the **Save Results As** button and select a location.
- If you specified **VCF** in the output popup (discussed in the second bullet point, above), or select **VCF** from the **Format** drop-down menu, you must provide a sequence annotated by SeqNinja with variants

from a VCF file. This is done in the row that appears under the **Save Results As** row. Check the **Use as template** box, then click Browse to navigate to the VCF file. The template may be used as a guide to the construction of the result file. For example, the meta-information lines may be copied from the template to the result file.

After setting up the template as desired, click **Run** to run the script and create a feature file. Or choose **Cancel** to exit from the template without running the script.

# Toolkit panel

---

The Toolkit panel provides four expand bars containing steps and operations to add to the script in the [Visual view](#). When you add a Toolkit item to an existing script, the tool operates on the selected sequence. When no selection is made, the step or operation is added to the bottom of the script.

To open the Toolkit panel:

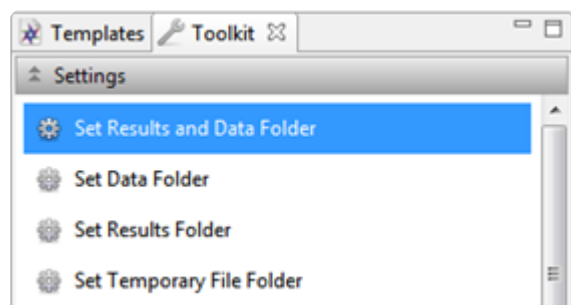
Use any of the following methods:

- Click on the **Toolkit** tab to bring the panel to the front.
- Choose **View > Toolkit**.
- Press **Ctrl+Alt+K** (Win) or **Cmd+Option+K** (Mac).

For information on opening, closing, minimizing, or maximizing the panel or its expand bars, see [Hide, show and change the sizes of items in the SeqNinja interface](#).

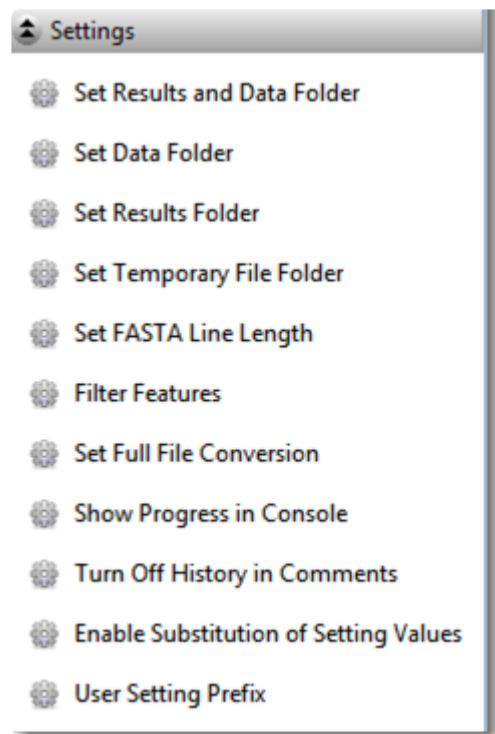
To insert a tool from the toolkit anywhere in an existing SeqNinja script:

[Drag & drop](#) it from the Templates panel to the [Visual view](#).



# Settings section

The Settings section is accessed via the upper-most of four expand bars in the [Toolkit panel](#). To open the section, click on the expand bar entitled “Settings.” If the bar isn’t visible, see [Hide, show and change the sizes of items in the SeqNinja interface](#).



Note that Settings items may be disabled depending upon the context of the current script and selection.

To add a Setting to the bottom of a script click on the name of the Setting in the Toolkit.


To add a Setting anywhere in a script [drag & drop](#) the Setting in the desired location. Alternatively, insert the cursor anywhere in the script, then click the Setting to apply it after the selected statement.

Each Setting uses a nearly identical format. The drop-down menu on the left is pre-selected for the Setting you chose. The text box on the right can be used to specify file paths, feature types, setting values, etc.



See the table below for information about each item in the Settings section:

Setting name	Pre-populated menu selection	Enter this information on the right	Item function
--------------	------------------------------	-------------------------------------	---------------

<b>Set Results and Data Folder</b>	datapath		To set the root directory for relative file paths used in the rest of the script.
<b>Set Data Folder</b>	datapath:in	Click on the file folder icon (  ) to specify the file path.	To set the root directory for input files. Using different directories for input and output reduces risk of accidentally overwriting an input file.
<b>Set Results Folder</b>	datapath:out		To set the root directory for output files. Using different directories for input and output reduces risk of accidentally overwriting an input file.
<b>Set Temporary File Folder</b>	tempdir		To specify a temporary files directory. This can be used to put temporary files on a drive with more space, or in a more easily accessible location.
<b>Set FASTA Line Length</b>	write:fasta:line	Type in a line length (integer)	To specify the line length for each sequence row in a FASTA-formatted output file. By default, FASTA sequences are broken into lines 60 characters long.
<b>Filter Features</b>	features	Type in text (e.g., <i>CDS</i> ).	To limit output features to those listed.
<b>Set Full File Conversion</b>	conversion	Type in <i>natural</i> or <i>full</i> .	When converting to a simpler file format, feature information can be preserved by using “full”. For example, when the input source is a Genbank or Lasergene file and the output file is in FASTA format, any features would normally be lost, since features are not supported in FASTA format. By changing the setting to full, SeqNinja creates auxiliary file(s) to preserve the features and comments. See <a href="#">Converting Between Annotated and Featureless Formats</a> for more information.
<b>Show Progress in Console</b>	verbose	Leave at default ( <i>TRACE, DATA</i> ).	To set verbose output of the types specified in the value. This can be useful for providing progress information and for trouble-shooting.
<b>Turn Off History in Comments</b>	comments:history	Type in <i>off</i> or <i>on</i> .	To control whether SeqNinja-related file history is added to the comments section of output files. If this setting is turned on, as it is by default, a comment such as “Sequence generated by SeqNinja” is added to the comments section of the output file(s).
<b>Enable Substitution of Settings Values</b>	setting:substitution	Type in <i>true</i> or <i>false</i> (default)	To enable/disable the substitution of settings in strings and integral expressions. This option is turned off by default. If you want user settings to be recognized in the script, you must turn the option on by adding this setting

			and changing its value to 'true.'
<b>User Setting Prefix</b>	user:	Type in a setting (a value, path, etc.). Also consider adding text after the colon on the left (e.g. <i>user:fasta</i> ) to differentiate the setting from other custom settings.	To create a custom user setting (a value, path, etc.) that can be used within a script. See <a href="#">Use Setting Substitution</a> for detailed information about this option.



# Converting between annotated and featureless formats

The [Set Full File Conversion](#) default setting of *natural* causes SeqNinja to write/read only those items that are “natural” to the receiving/source file type. Changing the setting to *full* when changing from an annotated format to a featureless format and back again creates auxiliary file(s) to preserve features and comments that would otherwise be lost.

✿ **Note:** The conversion setting must appear in the script before the operation in which file output is specified.

Converting from GenBank and Lasergene (.seq) format to FASTA format:

“Full” can preserve features and comments that would normally be lost when converting from annotated format to FASTA format, then back again. However, “full” is not suggested in all cases; for instance, it is not recommended when you plan to make sequence modifications (other than substitutions) while the sequence is in FASTA format.

When “full” is selected, SeqNinja outputs three files during conversion from an annotated format to FASTA format:

- *.fasta* – This is a standard FASTA file containing sequence reads. When writing FASTA files from a source that contains comments, the FASTA header line includes the first line of comments from the source. For example:  
  

```
> NC_010473 LOCUS NC_010473 4686137 bp DNA circular BCT 04-OCT-2012
```
- *.fasta.starff* – This file contains tab-delimited feature information from the original annotated file and can be opened and modified using a spreadsheet utility such as Microsoft Excel®. The modified file can be read by SeqNinja, with the following restrictions: 1) Sequences must appear in exactly the same order as in the corresponding FASTA file. 2) The first six columns must remain present in the original order. Subsequent columns contain qualifier values named in the column header, and can be removed or amended, as desired.
- *.fasta.comments* – This file contains one comment pertaining to each sequence in the FASTA file, and in the same order.

✿ **Note:** Comments are maintained during file conversions (e.g., a.seq = b.gbk). However, comments are discarded during sequence transformations (e.g., a.gbk = b.gbk + c.gbk).

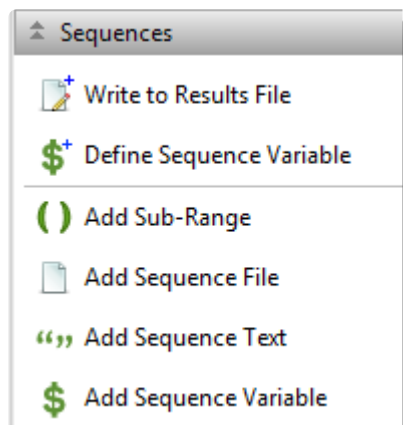
### Converting the saved FASTA file back to GenBank or Lasergene format:

If you used *full* during the conversion to FASTA, you could convert back to GenBank or Lasergene, using *full* again, without losing the original features and comments. As long as you keep the auxiliary files in the same folder as the FASTA data file, SeqNinja will automatically place the features and comments back in the GenBank or Lasergene file.

# Sequences section

---

The Sequences section is accessed via the second of four expand bars in the [Toolkit panel](#). To open the section, click on the expand bar entitled “Sequences.” If the bar isn’t visible, see [Hide, show and change the sizes of items in the SeqNinja interface](#).



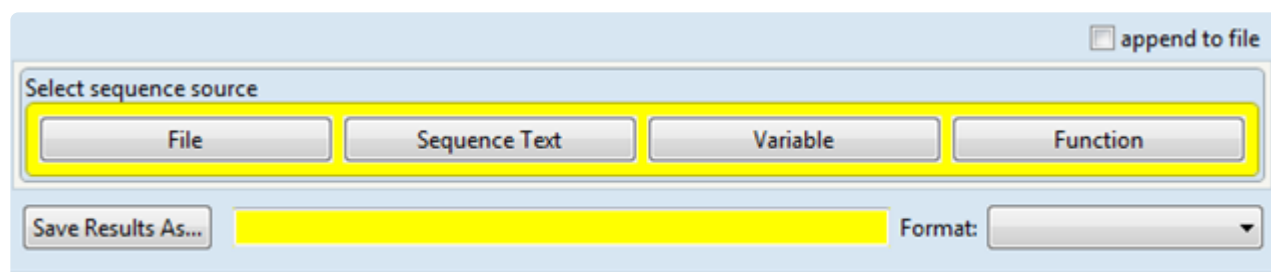
Note that Sequences items may be disabled depending upon the context of the current script and selection.

- To add a Sequences item to the bottom of a script, click on the name of the Sequences item in the Toolkit.
- To add a Sequences item anywhere in a script, [drag & drop](#) the Sequences item in the desired location. Alternatively, insert the cursor anywhere in the script, then click the setting to apply it after the selected statement.

Click any of the links below to learn more about items that can be applied from the Sequences section.

# Write to Results File

The Write to Results File tool, located in the [Sequences section](#) of the [Toolkit panel](#), allows you to add the following step to a script.

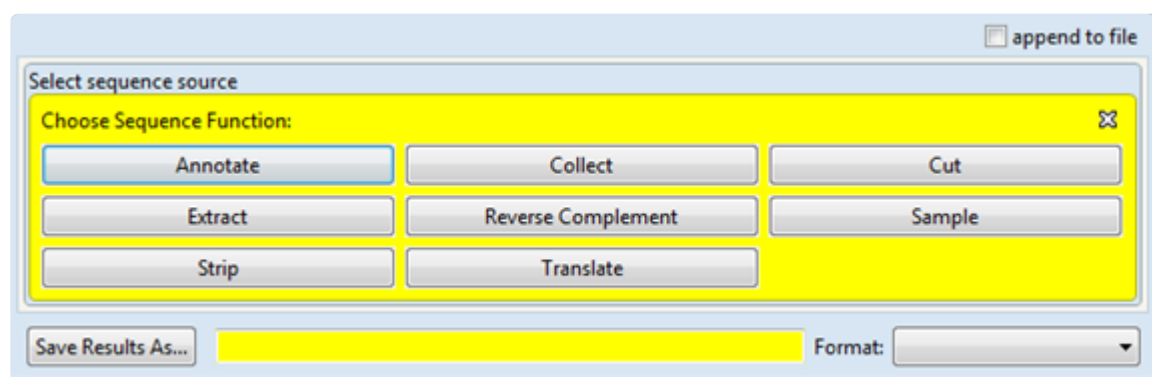


The dialog requires you to select a sequence source by clicking one of the four buttons: **File**, **Sequence Text**, **Variable** and **Function**. After pressing a button, the top of the dialog will be replaced with a version corresponding to your selection.

- If you selected **File**, input the desired sequence in the **Choose Sequence(s)** button row (see [Add and modify a sequence](#)). You may write to both sequence and feature files.
- If you selected **Sequence Text**, type any allowable text (letters or spaces) for the sequence into the yellow text box.
- If you selected **Variable**, type the desired sequence variable into the **Choose Sequence Variable** text box. Legal characters in a variable name are A-Z, a-z, 0-9, and \_ (underscore).

In the **Save Results As** area, choose the name, location and format in which to save the output (see [Specify output format and location](#)).

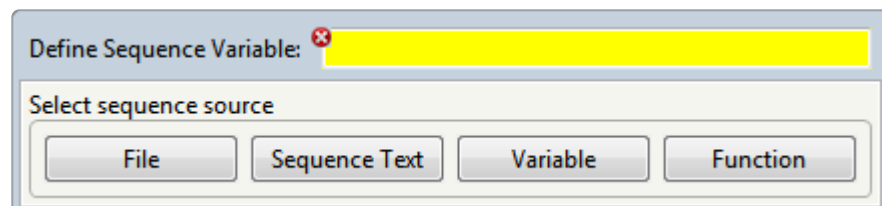
If you select the **Function** button, the following dialog replaces the original one. Click any of the function buttons, which open dialogs corresponding to matching items in the [Functions section](#).



# Define Sequence Variable

---

The Define Sequence Variable tool, located in the [Sequences section](#) of the [Toolkit panel](#), allows you to define a variable to represent any sequence expression such as a file, sequence text, another variable or a function.

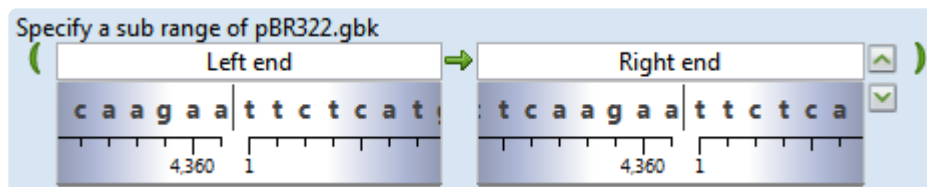


- In the **Define Sequence Variable** area, type the desired variable into the text box. Legal characters in a variable name are A-Z, a-z, 0-9, and \_ (underscore).
- This dialog requires you to select a sequence source. See [Write to Results File](#) for information on using the four buttons: **File**, **Sequence Text**, **Variable** and **Function**.

# Add Sub-Range

The Add Sub-Range tool, located in the [Sequences section](#) of the [Toolkit panel](#), can be added to a sequence that is already in the [Visual view](#) whether the sequence already has an existing range specified or not.

If this tool is added to a sequence without a range, it sets the ends of the sequence (i.e. specifies the first subsequence). If the tool is added to a sequence that already has a range specified, it adds a new subsequence.



- Input the desired range in the **Specify a range of the sequence** area (see [Set sequence ends](#)).

# Add Sequence File

---

The Add Sequence File tool, located in the [Sequences section](#) of the [Toolkit panel](#), allows you to add a sequence file to an existing script.

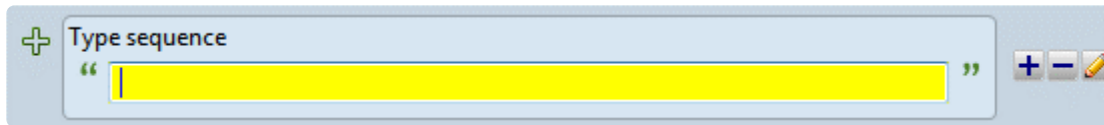


- In the **Choose Sequence(s)** button row, add another sequence to the dialog (see [Add and modify a sequence](#)).

# Add Sequence Text

---

The Add Sequence Text tool, located in the [Sequences section](#) of the [Toolkit panel](#), allows you to add sequence text to an existing script.



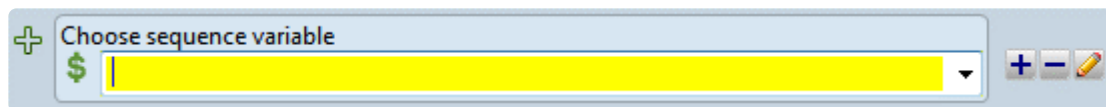
- In the **Type sequence** text box, type the insertion. Only nucleotides and amino acids are allowed.



# Add Sequence Variable

---

The Add Sequence Variable tool, located in the [Sequences section](#) of the [Toolkit panel](#), allows you to add a sequence variable to an existing script.



- In the **Choose Sequence Variable** area, use the drop-down menu to choose the variable, or type the desired sequence variable into the text box. You may only choose a variable that has been previously defined using the [Define Sequence Variable](#) tool.

# Functions section

The Functions setting is accessed via the third of four expand bars in the [Toolkit panel](#) and allows you to add a function to an existing script. To open the section, click on the expand bar entitled “Functions.” If the bar isn’t visible, see [Hide, show and change the sizes of items in the SeqNinja interface](#).



Note that some Functions items may be disabled depending upon the context of the current script and selection.

- To add Function to the bottom of a script, click on the name of the Function in the Toolkit.
- To add a Function anywhere in a script, [drag & drop](#) the Function in the desired location. Alternatively, insert the cursor anywhere in the script, then click the Setting to apply it after the selected statement.

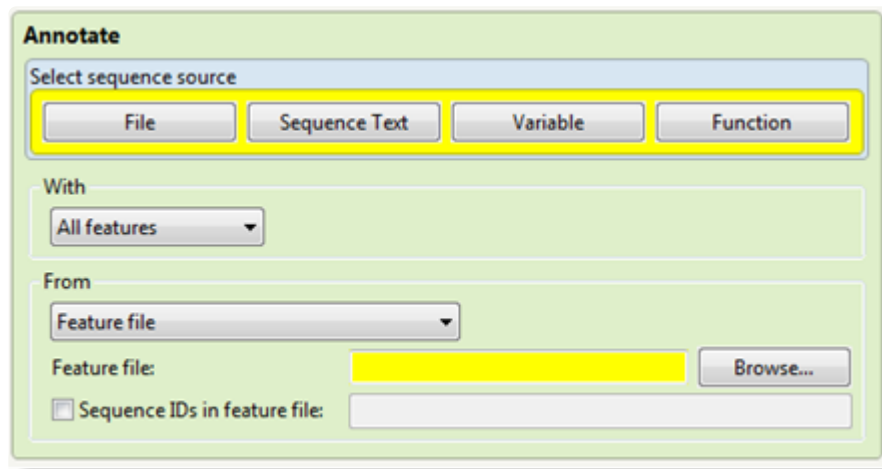
Functions dialogs are identifiable by their light green frames.

Click any of the links below to learn more about items that can be applied from the Functions section.

# Annotate

The Annotate function, located in the [Functions section](#) of the [Toolkit panel](#), adds a new annotation function to an existing script, allowing you to add features to a sequence. See its sister template topic, [Add Features](#), for detailed information and use cases.

Unlike the Add Features template, which prompts you to add a file, the Annotate function allows you to specify a sequence source using one of four buttons: **File**, **Sequence Text**, **Variable** and **Function**. See [Write to Results File](#) for more information on using these buttons.



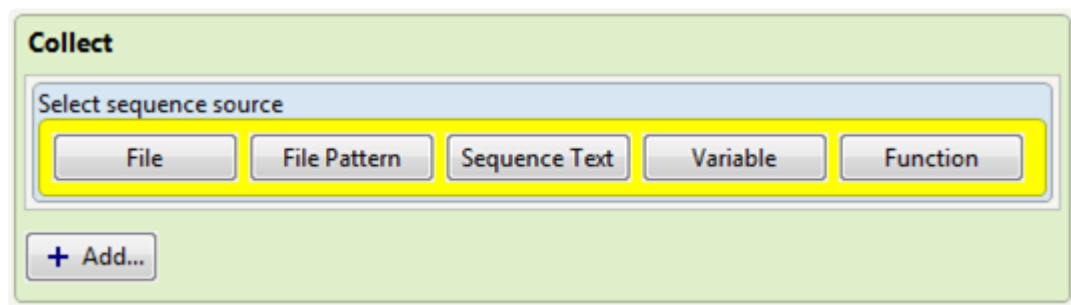
The screenshot shows the 'Annotate' dialog box with a light green background. At the top, the title 'Annotate' is in bold. Below it is a section titled 'Select sequence source' with a light blue header. This section contains four buttons: 'File', 'Sequence Text', 'Variable', and 'Function'. The 'File' button is highlighted with a yellow border. Below this section is a 'With' label followed by a dropdown menu showing 'All features'. Underneath is a 'From' label followed by a dropdown menu showing 'Feature file'. Below the 'From' dropdown is a 'Feature file:' label, a yellow text input field, and a 'Browse...' button. At the bottom, there is a checkbox labeled 'Sequence IDs in feature file:' followed by an empty text input field.

# Collect Sequences

---

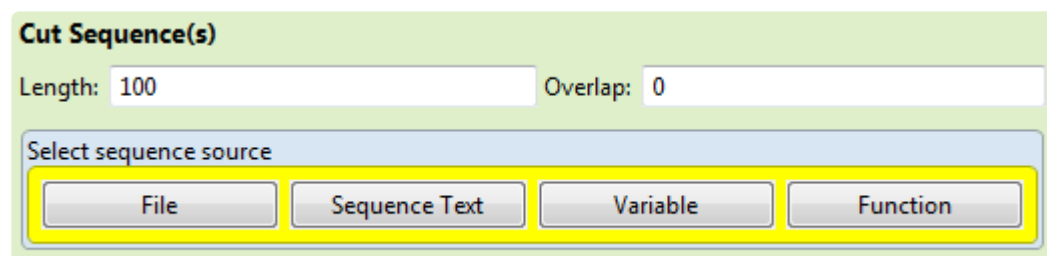
The Collect Sequences function, located in the [Functions section](#) of the [Toolkit panel](#), allows you to save multiple items (such as sequence files) into a single file. See its sister template topic [Collect Sequences Function](#) for detailed information and use cases.

Unlike the its sister template, this function requires you to select a sequence source using one of five buttons: **File Pattern** (described in [Use file patterns](#)), **File**, **Sequence Text**, **Variable** and **Function** (described in [Write to Results File](#)). By contrast, the template version prompts you to enter a file pattern by default.



# Cut Sequence(s)

The Cut Sequence(s) function, located in the [Functions section](#) of the [Toolkit panel](#), adds a new sequence cutting function to an existing script. This function allows you to cut sequences into pieces of a specified length. Cut Sequences is useful if you would like to break up a long sequence in order to perform a simulated assembly using software like SeqMan Ultra or SeqMan NGen.



**Cut Sequence(s)**

Length: 100      Overlap: 0

Select sequence source

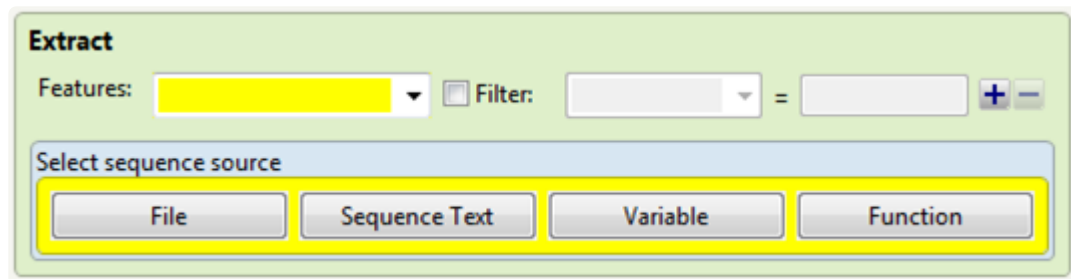
File      Sequence Text      Variable      Function

- Choose the desired **Length** of the sequence segments and the amount of **Overlap** to include in each segment.
- This dialog requires you to select a sequence source. See [Write to Results File](#) for information on using the four buttons: **File**, **Sequence Text**, **Variable** and **Function**.

# Extract Features Function

The Extract Features function, located in the [Functions section](#) of the [Toolkit panel](#), adds a new feature extraction function to an existing script. See its sister template topic, [Extract Features as Sequences](#), for detailed information and use cases.

Unlike its sister template, which prompts you to add a file, the Extract Features function allows you to specify a sequence source using one of four buttons: **File**, **Sequence Text**, **Variable** and **Function**. See [Write to Results File](#) for more information on using these buttons.

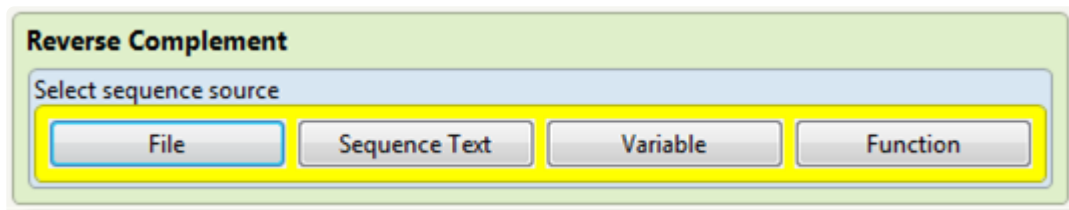


# Reverse Complement

---

The Reverse Complement function, located in the [Functions section](#) of the [Toolkit panel](#), adds a new reverse complement function to an existing script. See its sister template topic, [Reverse Complement Sequence](#), for detailed information and use cases.

Unlike its sister template, which prompts you to add a file, the Reverse Complement function allows you to specify a sequence source using one of four buttons: **File**, **Sequence Text**, **Variable** and **Function**. See [Write to Results File](#) for more information on using these buttons.

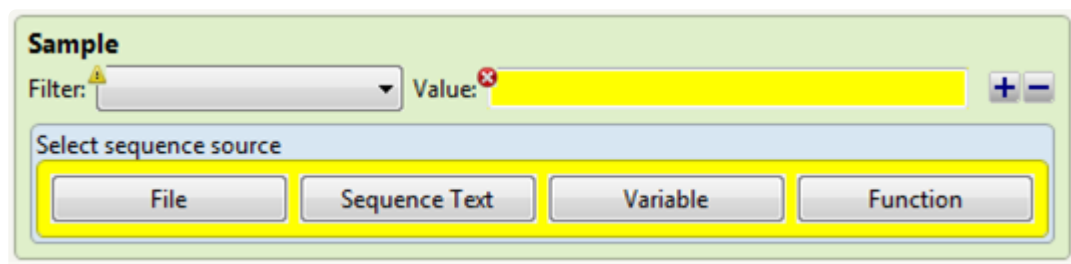


# Sample Sequences

The Sample Sequences function, located in the [Functions section](#) of the [Toolkit panel](#), allows you to sample the sequences in the input set. This can be useful for separating reads into different sets, or for reducing a very large number of reads to a smaller number (due to software limitations, for example). The output sequences appear in the same order in which they appeared in the original set.

For detailed information and use cases, see the help topic for this function's sister template, [Sample Sequences](#).

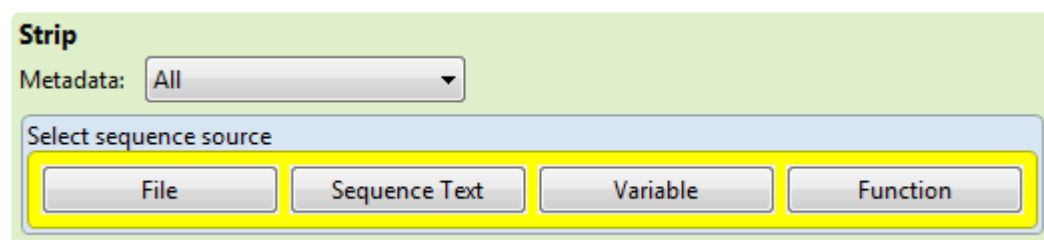
Unlike the sister template, which prompts you to add a file, the Reverse Complement function allows you to specify a sequence source using one of four buttons: **File**, **Sequence Text**, **Variable** and **Function**. See [Write to Results File](#) for more information on using these buttons.





# Strip Metadata

The Strip Metadata function, located in the [Functions section](#) of the [Toolkit panel](#), adds a metadata removal function to an existing script. This function may be used to remove specified annotations from a sequence.



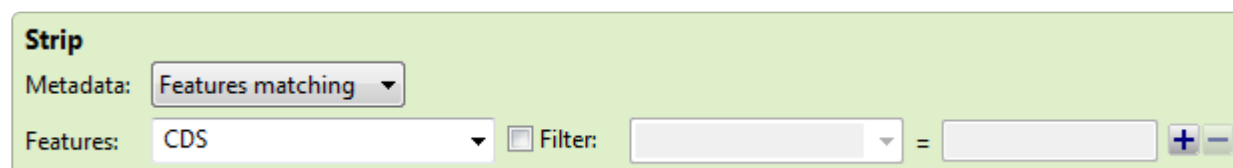
**Strip**

Metadata: All

Select sequence source

File Sequence Text Variable Function

- Use the **Metadata** drop-down menu to choose which annotations to remove: **All**, **Header** (to leave features and remove only the header text), **Features** (to leave the header and remove all features), **Features matching** or **Features except**. If you choose one of the latter two options, a new row named “Features” opens just under the drop-down menu.



**Strip**

Metadata: Features matching

Features: CDS ☐ Filter:  =  + -

Use the drop-down menu to select the feature type to match (or *not match*): **gene**, **CDS**, **exon**, **intron**, **mRNA**, **tRNA**, **promoter** or **misc\_binding**. To specify more than one feature type, use the plus (+) button to add additional “Features” rows. If you want to further limit the search to features matching (or *not matching*) particular qualifiers, check the **Filter** box, then choose a qualifier (**gene**, **/product**, **/locus\_tag**, **/note** or **/db\_xref**) from the drop-down menu to its right. In the right-most textbox, enter the text that the qualifier must match or *not match* (e.g., */gene = thrL*) in order for the feature to be removed in the output file. You may use wildcards in this box if you wish (e.g., */gene = thr\**).

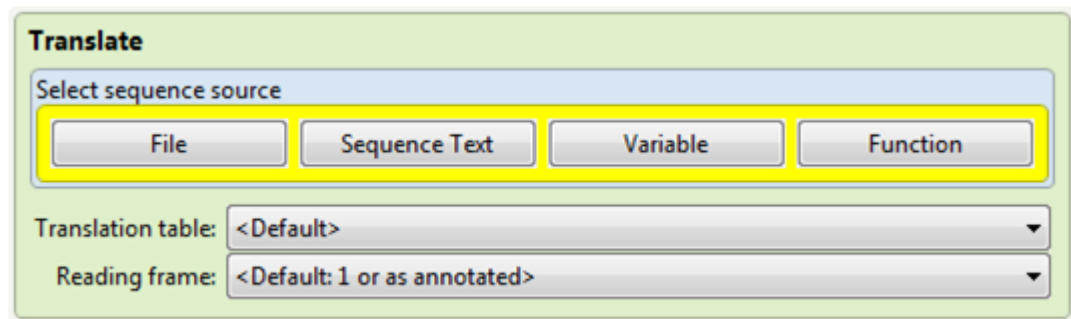
- The “Select sequence source” part of this dialog requires you to select a sequence source. See [Write to Results File xxx](#) for information on using the four buttons: **File**, **Sequence Text**, **Variable** and **Function**.

# Translate

---

The Translate function, located in the [Functions section](#) of the [Toolkit panel](#), adds a new DNA-to-protein translation function to an existing script. See its sister template topic, [Translate](#), for detailed information and use cases.

Unlike the Translate template, which prompts you to add a file, the Translate function allows you to specify a sequence source using one of four buttons: **File**, **Sequence Text**, **Variable** and **Function**. See [Write to Results File](#) for more information on using these buttons.



The screenshot shows a dialog box titled "Translate" with a light green background. Inside, there is a section titled "Select sequence source" with a yellow border containing four buttons: "File", "Sequence Text", "Variable", and "Function". Below this, there are two dropdown menus: "Translation table:" with the selected value "<Default>" and "Reading frame:" with the selected value "<Default: 1 or as annotated>".

# Label as DNA / RNA / Protein

---

The Label as DNA, Label as RNA and Label as Protein tools, located in the [Functions section](#) of the [Toolkit panel](#), mark the sequences in a set as being DNA, RNA, or protein. This ability may be useful for sequences originating in formats where type is unspecified, such as FASTA files.

If you use any of these functions, the information will be added to the output file, where allowed (e.g., *.gbk*). The presence of sequence type information may affect the results of searching in an endpoint expression. For example, in DNA, “N” = anything, whereas in protein, “N” = asparagine.

# Label as Circular / Linear

---

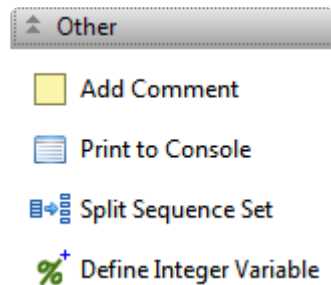
The Label as Circular and Label as Linear tools, located in the [Functions section](#) of the [Toolkit panel](#), mark the sequences in a set as being linear or circular. This ability may be useful for sequences originating in formats where type is unspecified, such as FASTA files.

If you use either of these functions, the information will be added to the output file, where allowed (e.g., *.gbk*). The presence of sequence type information may affect the results of searching in an endpoint expression. For example, a match can cross the origin in a circular sequence, but not in a linear one.

# Other Section

---

The Other section is accessed via the lower-most of four expand bars in the [Toolkit panel](#). This section allows you to type in any desired text (as a comment), define integer variables, etc. To open the Other section, click on the expand bar entitled “Other.” If the bar isn’t visible, see [Hide, show and change the sizes of items in the SeqNinja interface](#).



Note that some “Other” items may be disabled depending upon the context of the current script and selection.

- To add an “Other” item to the bottom of a script, click on the name of the “Other” item in the Toolkit.
- To add an “Other” item anywhere in a script, [drag & drop](#) the “Other” item in the desired location. Alternatively, insert the cursor anywhere in the script, then click the Setting to apply it after the selected statement.

Click any of the links below to learn more about items that can be applied from the Other section.

# Add Comment

---

The Add Comment tool, located in the [Other section](#) of the [Toolkit panel](#), allows you to add a typed comment to the script.



- In the yellow text box, type any desired text, including letters and numbers. Note that comments do not affect the running of the script.

# Print to Console

---

The Print to Console tool, located in the [Other section](#) of the [Toolkit panel](#), directs SeqNinja to display the selected sequences, strings or integer expressions in the [Console](#).



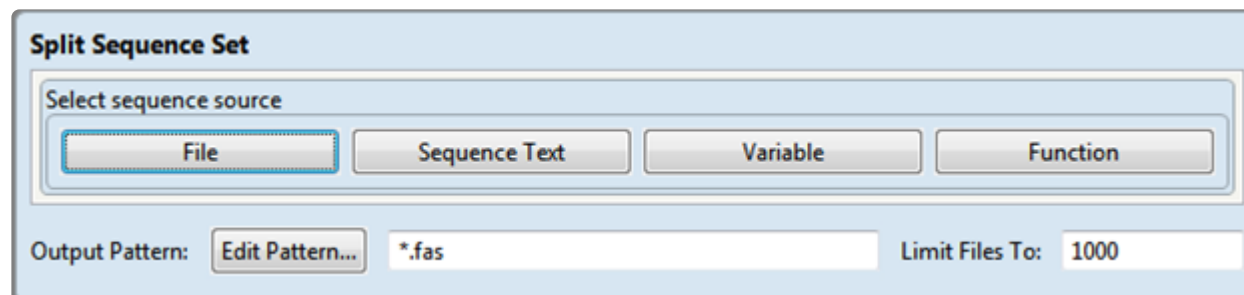
Type in the sequence, string, or integer expression you wish to display in the Console.

See the topic [Use escape codes](#) for an example that uses the Print to Console tool.

# Split Sequence Set

The Split Sequence Set tool, located in the [Other section](#) of the [Toolkit panel](#), separates multi-sequence files into multiple files, each consisting of a single sequence.

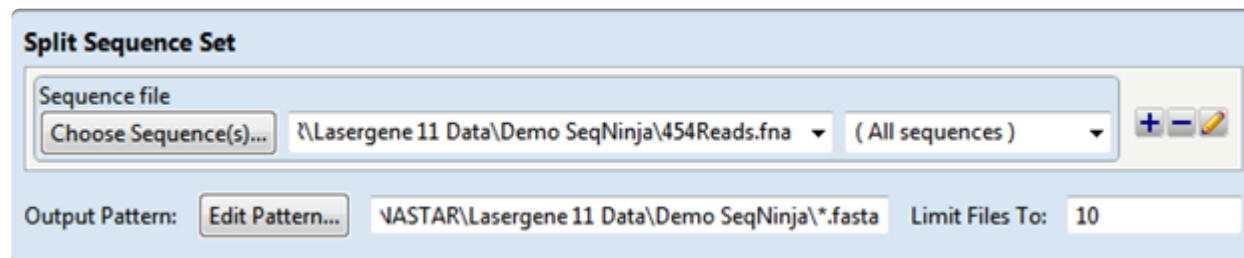
Note: Sequences that have been split may later be combined in a common file using the [Collect Sequences](#) tool.



- This dialog requires you to select a sequence source. See [Write to Results File](#) for information on using the **File**, **Sequence Text**, **Variable** and **Function** buttons.
- To specify the required filename pattern, click the **Edit Pattern** button. See [Use File Patterns](#) for detailed information.
- If desired, enter a value in the **Limit Files** To box to limit the number of sequence files that can be generated during a run. The default is 1000.

If you have specified “full” [file conversion](#), each sequence file that is generated may be accompanied by auxiliary files, as well.

## Example:



The input file, *454Reads.fna*, consists of 13 sequences in a multi-sequence file format.

After running the script, the first ten sequences in *454Reads.fna* are saved as individual sequence files in *.fasta* format.



# Define Integer Variable

---

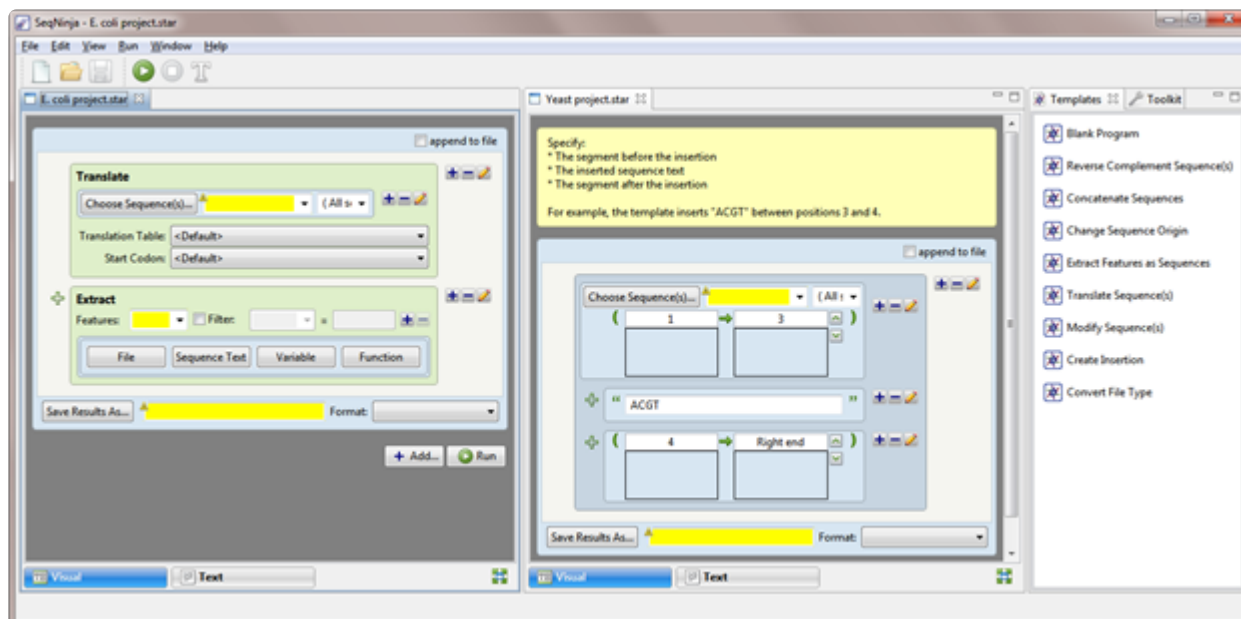
The Define Integer Variable tool, located in the [Other section](#) of the [Toolkit panel](#), allows you to add an integer variable to the script. This function is designed for advanced SeqNinja users.

The image shows a graphical user interface for the 'Define Integer Variable' tool. It consists of a horizontal rectangular box with a thin border. On the left side of the box is a green percentage symbol (%). To its right is a white text input field. Further right is a green equals sign (=). To the right of the equals sign is another white text input field, which is longer than the first one.

- To the left of the equals sign, type the desired sequence variable into the text box. Legal characters in a variable name are A-Z, a-z, 0-9, and \_ (underscore).
- To the right of the equals sign, type what the variable signifies.

# Split the SeqNinja screen





By default, only one script at a time is displayed, even if several scripts are open. To view two or more scripts simultaneously on a split screen, drag a script tab in a downward, right or left direction. When you see blue box outlines in the desired orientation (side-by-side or stacked), release the mouse button.



Scripts can be re-docked in their original positions by reversing this process.

# Hide, show and change the sizes of items in the SeqNinja interface

Controls for SeqNinja's views, panel, and expand bars are described in the following table. Not all options in the table are applicable to every item. Note that optimizations are applied only to the active project and do not affect any other open SeqNinja projects.

Task	How To
<b>Return to the default SeqNinja display</b>	Use <b>View &gt; Reset Panel Layout</b> . Use of this command does not affect the state of expand bars.
<b>Show or open a hidden item</b>	Use the associated View menu command, click on a closed expand bar., or click on a plus sign.
<b>Hide or close a visible item</b>	Use the <b>Close</b> icon (  ) or <b>Minimize</b> icon (  ), click on an open expand bar, or click on a minus sign. To close one or more tabs, right-click on a project, view or panel tab and choose <b>Close</b> , <b>Close Others</b> , <b>Close Tabs to Right</b> , or <b>Close All</b> . Some of these options are only available for certain tabs.
<b>Increase or decrease the size of an item</b>	Drag the bottom or left sides of the item or use the <b>Maximize</b> icon (  ) or <b>Restore</b> icon (  ). To view more of an open expand bar, close neighboring bars and/or increase the size of the whole panel.
<b>Move an item</b>	Using the mouse, grab a project, view or panel tab, then <a href="#">drag and drop</a> it in the desired location.

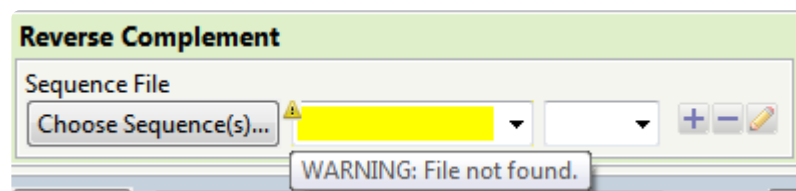
# Work with Sequences

---

- To start a new SeqNinja script in the [Visual view](#), use **File > New Script**.
- To open a SeqNinja (.star) file, use the **File > Open** command, press **Ctrl+O** (Win) or **Cmd+O** (Mac), or drag & drop the file onto an open SeqNinja window. Once a script is open in SeqNinja, you can open additional files using the same command. See our [Supported File Types](#) page for a complete list of supported extensions.
- To open a new SeqNinja window with no scripts loaded, choose **Window > New Window**.

# Add and modify a sequence

Many of the [Templates panel](#) and [Toolkit panel](#) items require you to choose a sequence file.



- Click the **Choose Sequence(s)** button to open a file chooser. Alternatively, you may type in the path and file name in the drop-down menu (or text box) to the right of that button. The sequence you enter will apply only within that step and not to subsequent steps.
- Use the plus and minus icons on the right to either add another sequence or to delete an existing sequence (see table, below).

## \* Notes:




**About the yellow “warnings”:** Until you choose a valid sequence, the drop-down menu (or text box) will be highlighted in yellow and may also display a triangular warning icon. Hovering over the sequence name displays a balloon tip: “WARNING: file not found.” This warning is normal and will disappear once you choose a valid sequence.

**About the use of quotation marks in paths:** If you are using the [Text view](#), double-quotes should be used around most paths, whether in settings or sequence expressions. Double-quotes inside of paths, however, are not supported. When using the [Visual view](#), use paths that do not contain double-quotes. Also, for [File Pattern](#) specification, single-quotes must be used to distinguish file patterns from individual file paths, which are double-quoted.

**About “dependent” sequences:** Within the same script, you may use the output sequence from an earlier step as the input sequence for a later step. However, the result might not be what you expected or desired. For example, if an extracted CDS initially had a /codon\_start annotation (i.e., it did not start with an intact codon), that information would be lost in the extracted FASTA file. Then, when the file was subsequently translated, it would be in the wrong reading frame.

- The right-most drop-down menu, which is initially set to “(All sequences),” lets you further narrow down the sequence of interest by choosing one individual sequence from the selected file. Only the first 100 sequence names are displayed. To display additional sequences, click < **more** > at the bottom of the menu to bring up a chooser dialog. If no selection is made from the menu, the command will be executed on the entire file of sequences rather than on one individual sequence. Exception: when a sub-range or concatenation uses the file, and no selection is made, only the first sequence is used.


- The following tools are available to the right of any **Choose Sequence(s)** button row:

Tool	Function	Description
	<b>Add</b>	This button offers a sub-set of the “Add” items from the <a href="#">Sequences section</a> .
	<b>Remove</b>	Depending on the operation, clicking this button offers one or more of the following choices: <b>Remove</b> , <b>Delete</b> , <b>Undo</b> .
	<b>Modify</b>	The pencil icon lets you modify an operation by nesting the current operation into a newly-selected one. The function modifies the sequence directly to the left of the icon. Depending on the operation, clicking this button lets you modify the selected item using shortcuts to <a href="#">Functions section</a> items.

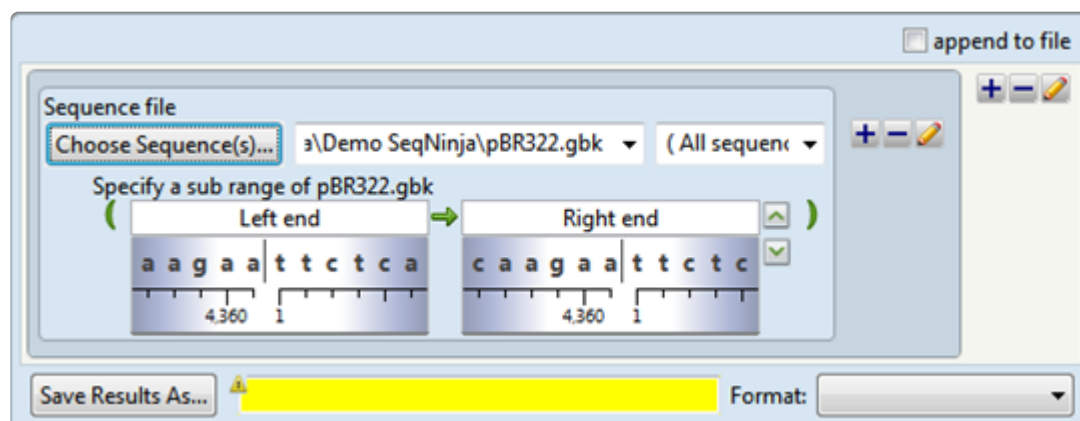
- For right-click menu options available in the **Choose Sequence(s)** button row, see the last table in Right-click functionality in the [Visual view](#).

# Set sequence ends

For any function that allows you to choose a sequence, you can also choose to set ends for the sequence. There are four ways to choose a sub-range for a sequence in the [Visual view](#):

- Via the [Modify Sequence](#) template.
- From a row with a **Choose Sequence(s)** button, right-click near the button and choose **Add Sub-Range**.
- From a row with a **Choose Sequence(s)** button, click on the plus icon (  ) to the right of the row, then choose **Add Sub-Range**.
- From the [Sequences section](#), apply [Add Sub-Range](#) by clicking on it or by drag-and-dropping it on the desired location in the Visual view.

The sequence range text boxes are initially populated with the terms “Left end” and “Right end.”



The following table shows tasks that can be done from within this dialog, or that affect its appearance.

Task	How To
To reveal/hide the sequence, ruler and any features that are present; and to set ends based on features or by direct selection	Click the up/down arrows to the right of the range text boxes. See <a href="#">Work with Features</a> for detailed instructions on how to view and work with sequence features.
To specify the desired range	Use the “Left end” and “Right end” boxes. You may type in a position or search for strings such as “atgc”. Alternatively, drag the “wheel” to go to a particular position number. Click any base to center the wheel on that base. You can reference the “left” of a sequence in the right-hand box, if desired (e.g., “left + 12”).

<b>To skip to the left or right ends of the sequence</b>	Click in a range text box, then use the left/right arrows that appear within the box.
<b>To view feature information</b>	Hover above a feature (if any are present) to see information about it, such as its left and right coordinates.
<b>To reverse complement the current sub-range</b>	Click the green arrow between the two range text boxes.

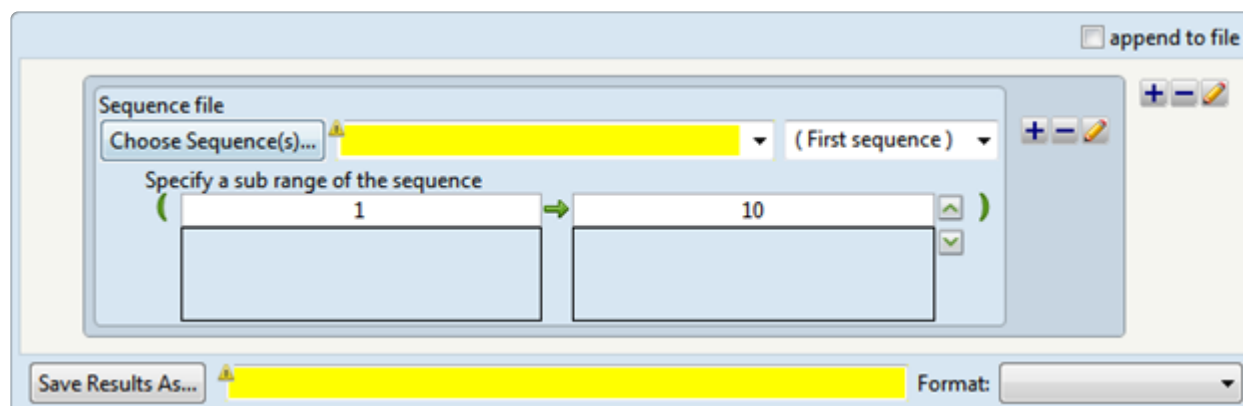
IUPAC ambiguity codes are recognized both in the sequence and in the range boxes. For example, typing AAS (where S = C or G) into a range box would cause SeqNinja to look for the first instance of AAC or AAG in the sequence. Conversely, typing AAC into a range box would cause SeqNinja to look for the first instance of AAC, as well as any combination of bases and ambiguity codes that would allow for AAC (e.g., AAS, WWM, etc.).



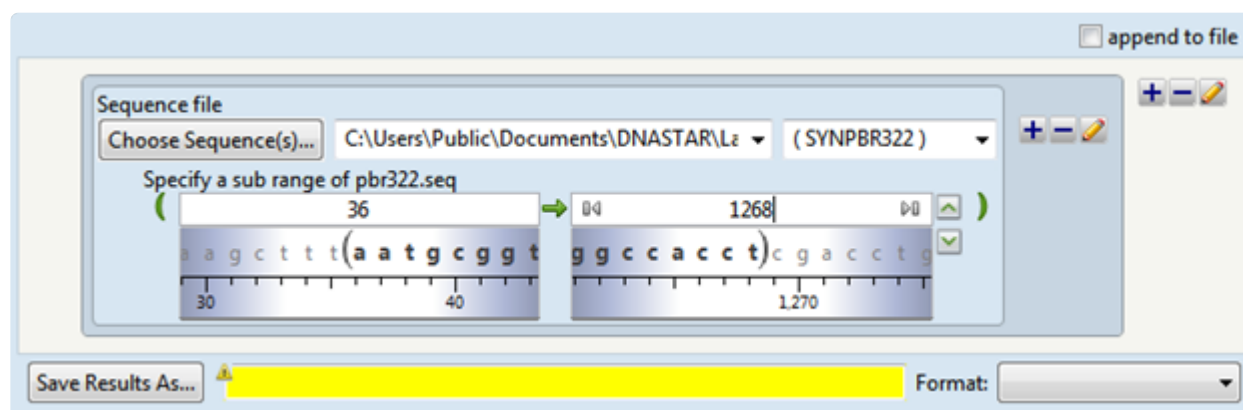
# Work with Features


SeqNinja lets you view sequence annotations from any [sub-range](#) step in the [Visual view](#). You can also set ends automatically based on one or more features.

Start by creating a sub-range step using any of the methods described in [Set sequence ends](#). The step initially appears as follows:



Next, choose a sequence and enter the desired sub-range. Two shaded “wheels” appear under the range boxes and display the sequence surrounding each coordinate.

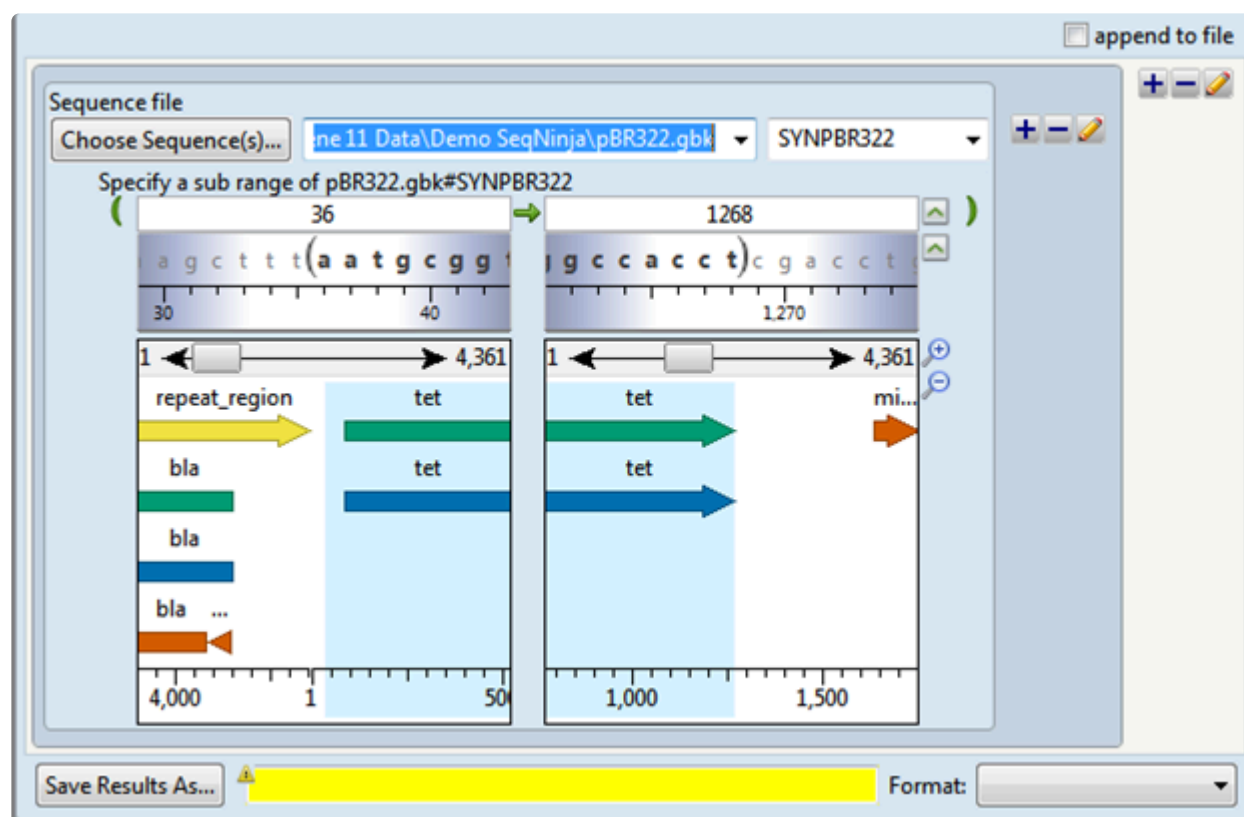


To view features, click the downward-pointing chevron to the far right of the range boxes (). Features appear below the wheels. If you are sure that your sequence contains features, but you do not see any after clicking the green arrow, use one of the methods in the table below to scroll along the sequence.

✿ **Note:** The display of features can be affected by the features and conversion commands. For example, if there is a step above the sub-range step in which you set **features = CDS**, the sub-range features display below that step will show only CDS features. To set endpoints based on features that you plan to later filter from the output, temporarily set **features = \***. After setting the endpoints, change the features setting to the type that you actually want to use for the output.

The sequence and feature displays treat all sequences as “circular,” in that both displays allow you to scroll “360°” around a sequence. By contrast, range boxes treat all nucleic acid sequences as circular, and all protein sequences as linear.

✿ **Note:** SeqNinja will return a red warning icon (⚠) if you enter a coordinate greater than the sequence length of a protein, but not of a nucleic acid. If you run a script that shows the warning icon, you will receive an error message with the statement: “[Sequence endpoint is greater than length of a linear sequence.]”


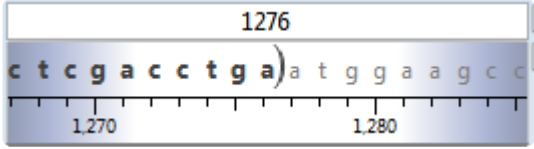
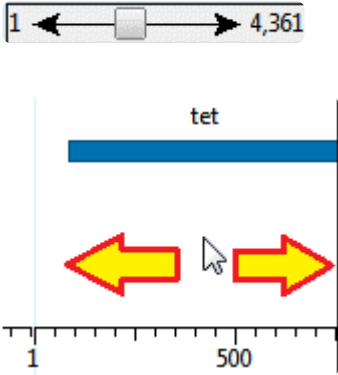
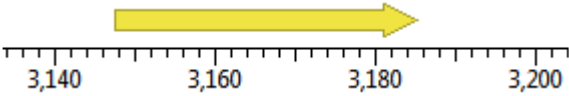



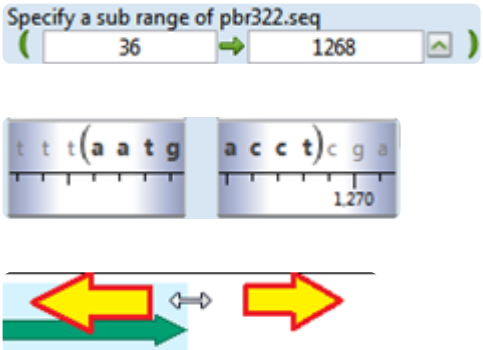
Each feature is displayed as a colorful arrow with the name of the feature displayed above. A rightward-pointing arrow indicates that the feature is on the current strand, while a leftward-pointing arrow signifies that the feature is on the opposite strand.

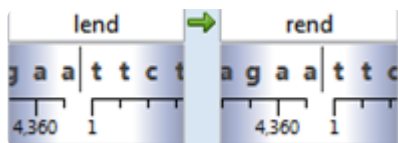
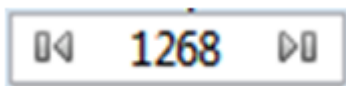
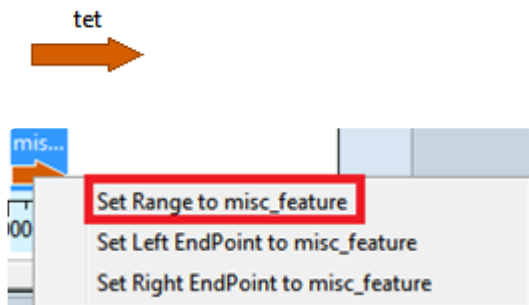
Light-blue highlighting is used to denote the currently selected range of the sequence.

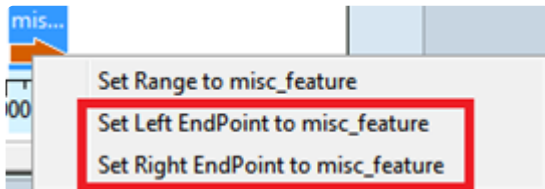
The following table shows tasks that affect the view or selection:

Task	How To	Images
------	--------	--------

<b>To display or hide the sequence and/or features</b>	<p>Click on the green chevrons to the far right of the range boxes. Hover over each chevron to view a tool tip showing the result that will occur if you click that chevron.</p>	
<b>To view the bases bracketing the current range box coordinate</b>	<p>Consult the shaded wheel just below the range box.</p>	
<b>To view other sections of the sequence without changing the range coordinates</b>	<p>Use the slider under each shaded wheel.</p> <p>Or...</p> <p>Drag the white features area background left or right.</p>	
<b>To view the range currently visible in the feature display</b>	<p>Consult the ruler under the feature display.</p>	

<p><b>To display a tooltip with available information about a feature</b></p>	<p>Use the mouse to hover over the feature.</p>	
<p><b>To switch the direction of the sequence</b></p>	<p>Click the green arrow between the two range boxes.</p>	
<p><b>To cause the sequence direction to match the direction of a particular feature.</b></p>	<p>Double-click on the feature. Double-clicking a left-facing feature will always flip the sequence direction.</p>	
<p><b>To zoom in or out on the features</b></p>	<p>Use the <b>Zoom In</b> and <b>Zoom Out</b> tools to the right of the feature display.</p>	
<p><b>To change the range coordinates via the range boxes, sequence or features display</b></p>	<p>Type a new coordinate into one or both range boxes.</p> <p>Or...</p> <p>Drag the shaded wheels under the range boxes left or right.</p> <p>Or...</p>	

	<p>Grab either end of the current (blue-shaded) selection in the features area and drag it to the left or right.</p>	
<p><b>To change the range coordinates to match the left or right end coordinates</b></p>	<p>Type “lend” or “left end” (for left coordinate) or “rend” or “right end” (for right coordinate) into one or both range boxes.</p> <p>Or...</p> <p>Single-click in a range box to activate the <b>Go to left end</b> and <b>Go to right end</b> shortcuts. Then click on the desired shortcut to enter that coordinate in the range box.</p>	 
<p><b>To change the range coordinates to set the sub-range to match feature coordinates</b></p>	<p>Double-click on any feature. Note that double-clicking on a feature facing opposite to the direction of the green arrow ( or ) will change the sequence direction.</p>	

	<p>Or...</p> <p>Right-click on any feature and choose <b>Set Range To [feature_name]</b>.</p>	
<p><b>To change the range coordinates to set the sub-range to match coordinates across two or more features</b></p>	<p>Right click on the feature furthest to the left (or right) and select <b>Set Left (or Right) EndPoint to [feature_name]</b>. Then scroll to the feature on the opposite end of the range and choose the opposite right-click command.</p>	

# Work with Scripts

---

To learn about working with scripts, see the following topics:

- [Specify file paths](#)
- [Specify output format and location](#)
- [Perform multiple calculations](#)
- [Use file patterns](#)
- [Use escape codes](#)
- [Use setting substitution](#)
- [Use editing commands](#)
- [Create a script to use as MegAlign Pro input](#)
- [Run a script](#)
- [Stop running a script](#)
- [Access run results](#)
- [Use drag & drop](#)

# Specify file paths

---

DNASTAR software does not support backward slashes (\) in file paths for Macintosh. Therefore, when typing file paths in SeqNinja, we recommend always using forward slashes (/), which are supported on both Windows and Macintosh platforms.



# Specify output format and location

Many of the [Templates panel](#) and [Toolkit panel](#) items require you to choose a filename and location for output data.

- Press the **Save Results As** button to select a location to save the file. Alternatively, you may type a path and filename into the text box on the right. A third way to designate the “Save Results As” name is by double-clicking on the title bar.

## \* Notes:

**About the default file location:** If a location is not specified, the file will be stored in the default output directory, which is your home directory.

**About the use of quotation marks in paths:** If you are using the [Text view](#), double-quotes should be used around all paths, whether in settings or sequence expressions. Double-quotes inside of paths, however, are not supported. When using the [Visual view](#), use paths that do not contain double-quotes. Also, for [File Pattern](#) specification, single-quotes must be used to distinguish file patterns from individual file paths, which are double-quoted.

- Use the **Format** drop-down menu to select the format in which you wish to save the file.
- If a file with the same “Save Results As” name and location already exists, checking **append to file** will append the output to the existing file rather than overwriting it.

# Perform multiple calculations

---

When performing multiple calculations, the steps must be set up in a particular way. For example, if the input file for one step is the output file from an earlier step, you may obtain unexpected results. Specifically, the final output may have additional sequences and/or features.

Here is an example of a workflow that can lead to such unexpected results:

- Step 1: Extract features from TEST.GBK and save them as EXTRACTED.GBK.
- Step 2: Translate EXTRACTED.GBK and save the results as TRANSLATED.GBK.
- Result: The file TRANSLATED.GBK may contain unexpected data (e.g., an extra translated feature).

This situation arises when there are overlapping CDS features. In such cases, a piece of one CDS will end up being annotated in the interval it shares with the second CDS. Then, when the EXTRACTED.GBK file is translated, that particular sequence will result in two protein sequences: the desired full length CDS, and the fragment of the overlapping CDS. Note that this is not an issue when the intermediate file is in FASTA format since there is no carryover of information about the overlapping CDS.

# Use file patterns

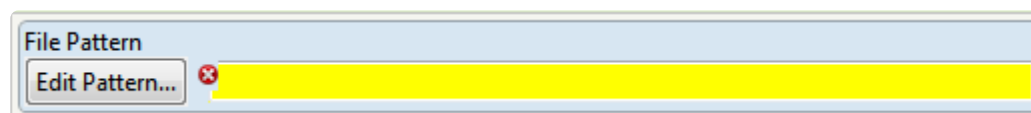
A file pattern is a search string with one or more wildcards in the filename. The [Collect Sequences](#) tool and the [Split Sequences](#) tool both support file patterns as arguments.

The following are some rules pertaining to the use of file patterns:

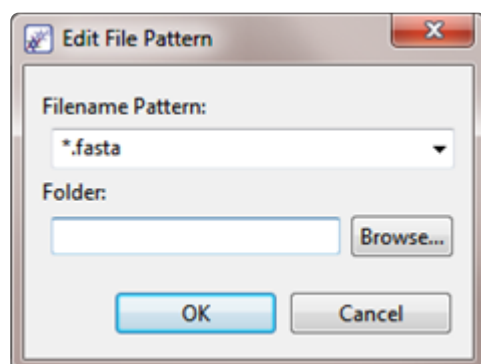
- A question mark (?) wildcard matches exactly one arbitrary character, and an asterisk (\*) wildcard matches zero or more arbitrary characters.
- A file pattern may be relative or absolute. Relative paths in patterns are resolved in the same manner as for double-quoted path names.
- Wildcards can only be used in the filename, and nowhere else in the path.
- Single-quotes must be used to distinguish file patterns from individual file paths, which are double-quoted. This makes it possible to reference files whose names contain wildcard characters.

When using the [Collect Sequences](#) tool, a file pattern is specified using the File Pattern button (see image below). Both the question mark (?) and asterisk (\*) wildcards are allowed.

When using the [Split Sequences](#) tool, a file pattern is specified in the Output Pattern row. Only the asterisk (\*) wildcard is allowed.



Click the **Edit Pattern** button to launch the Edit File Pattern dialog:



Use the **Filename Pattern** drop-down menu to choose from a variety of file extensions. The asterisk shown before a file extension will be replaced by the sequence name during execution of the script. In file patterns, the question mark (?) wildcard matches exactly one arbitrary character, and an asterisk (\*) wildcard matches zero or more arbitrary characters.

The following are examples of file patterns. The last row in the table shows how filename patterns may be combined to refer to filenames with multiple extensions.

Example	File Pattern
<i>.*fasta</i>	Uses all the files ending in <i>.fasta</i> in the default input directory.
<i>*.fas</i>	Uses all the files whose name contains <i>.fas</i> (e.g., <i>.fas</i> and <i>.fasta</i> files) in the default input directory.
<i>G*.fasta</i>	Uses all the files starting with <i>G</i> and ending with <i>.fasta</i> in the default input directory.
<i>Myfolder/*.fasta</i>	Uses all the files that end in <i>.fasta</i> that are located in the <i>Myfolder</i> subdirectory of the default input directory.
<i>C:/MyFolder/*.fasta</i>	Uses all the files in <i>C:/MyFolder</i> that end in <i>.fasta</i> .
<i>*.gb, *.genbank</i>	In the Collect Sequences step, use the <b>Add</b> button so there are two <b>File Pattern</b> rows. Uses all files ending in either <i>.gb</i> or <i>.genbank</i> in the default input directory.

- In the **Folder** area, type in the path to the desired output folder, or navigate to it using the **Browse** button.
- When you have finished with this dialog, click **OK**.

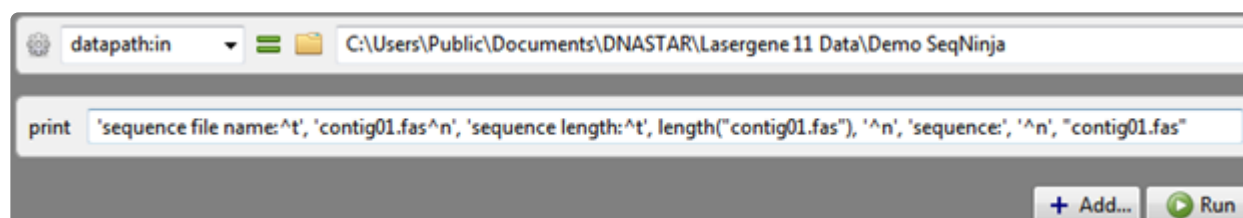
# Use escape codes

All quoted strings accept certain “escape codes.” These escape codes indicate that SeqNinja should put a tab, a new line, a quotation mark, etc. in the specified location:

Escape Code	Effect
<code>^t</code>	Inserts a tab.
<code>^n</code>	Inserts a line break.
<code>^^</code>	Inserts a caret character (^).
<code>^'</code>	Inserts a single quote into a string that is bracketed by single quotes.
<code>^"</code>	Inserts a double quote into a string that is bracketed by double quotes.

## Example:

This example, using the [Print to Console](#) tool, causes the name of the file, length of the sequence, and the sequence itself to print to the [Console panel](#) in a neatly organized manner.



The output in the Console panel is shown below:

```
sequence file name:    contig01.fas
sequence length:      116581
sequence:
TATGTCAGTAATTACCGCGTTCGCCAGCGTCAGTTCTCTGGCATT TTTGTCGCGCTGGGC
TTTGTAGGTAATGGCGTTATCACGGTAATGATTAAACAGCCCATGACAGGCAGACGATGAT
GCAGATAACCAGAGCGGAGATAATCGCGGTTACTCTGTTTATTGCTGACCCACAAACAG
ATTCACGCTCAATCTCACGACGAGTCATGAGACCTTCCATTGCTTACCGCCAGCATAT
GTCCAGCGACGTAGCTGATCATGCGCCTTTGATATCGCCCTGGTTTATTTTGC GAAGA
AGCGTCGATGTTCTGA AATTGCGAGCAGCCAGCTTGTAAACGAAATGAGTAAAGAGCGCCG
```

# Use setting substitution

SeqNinja allows you to substitute the values of settings into single- and double-quoted strings. You can substitute the value of any setting, including built-in settings, custom settings, and command-line arguments. When you run the script, SeqNinja will replace the name of the setting with its value. For example, a string (e.g., a file path) or a numeric setting (e.g., an integer) will be used in place of the setting name. This functionality allows the same script to be reused with different data and allows the script to be moved along with its data.

Substitution involves several steps, each of which is discussed below.

## Step 1 – Enabling substitution:

Before substitution is used in a script, you must first enable “substitution” in that script. In the [Visual view](#), this is easily done by adding the setting called **Enable Substitution of Setting Values** anywhere in the script above the substitution.



To enable substitution, the text box on the right must contain `true`, as shown in the image above. If it instead says `false`, overwrite the text with `true`.

✿ **Note:** The syntax for enabling substitution is `!setting:substitution = true`. This segment of the script can be observed by switching temporarily from the [Visual view](#) to the [Text view](#).

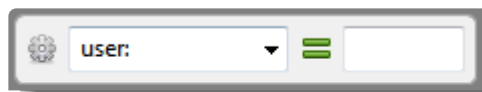
## Step 2 – Defining a setting:

Settings can be defined...

- in the script.
- as an argument typed into the Run with Options dialog.
- implicitly (see the “Tip,” below, for an example).

To define a setting in the script:

In an open script, click on the [Settings](#) tool **User Setting Prefix**. The following step is added to the script:



By default, the drop-down menu on the left is set to '@user@'. To differentiate this custom setting from other custom user: settings, you may type additional text to the right of the colon, if desired. For example, `user:tags`, `user:fasta`, `user:genbank`.

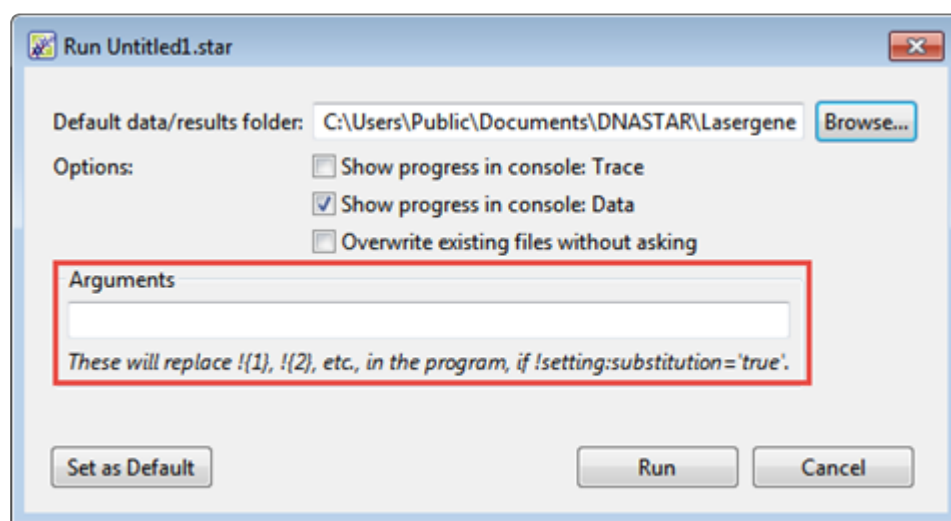
In the text box on the right, enter the information you want to substitute for the setting when the script is run (e.g., a file path, integer, etc.). Substitution can occur in numeric expressions or in strings. Strings are sequences of characters (e.g., a file path) between pairs of single or double quotes.

**Note:** The syntax for a user setting is `!user:<optional text> = <value>`. This segment of the script can be observed by switching temporarily from the Visual to the Text view.

To define a setting as an argument in the Run with Options dialog:

SeqNinja lets you define a custom setting as a script argument in the Run with Options dialog.

1. After setting up the script as desired, choose **Run > Run with Options**.



2. Type one or more arguments into the **Arguments** box. Multiple arguments should be separated by a space. Note that an argument can only define settings whose name is a positive integer.


**Example:** You set up a script that deletes the first 'n' bases of a sequence. The script uses `!${1}` to refer to the sequence file, and `!${2}` to specify the number of bases ('n'). In this example, you could use the arguments `myfile.fasta` for `!{1}` and `42` for `!{2}`. Therefore, you would enter the text `myfil`

e.fasta 42 into the **Arguments** box.

3) Reference the setting in the script using `!{1}`, `!{2}`, and so forth. If you are creating a script in the [Visual view](#), there is no specific step in which the setting is referenced. Simply type the substitution text into any appropriate text box.

### Step 3 – Specifying substitution:

Curly braces are used to substitute the value of a setting in a string or numeric expression.

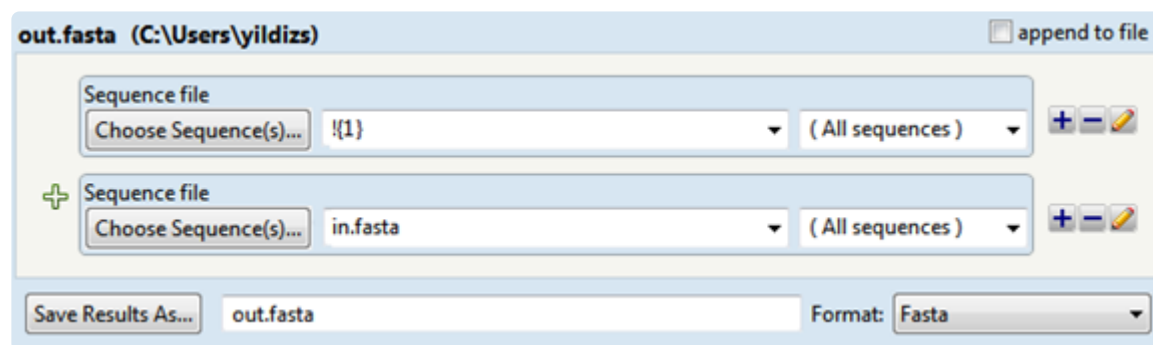
 **Tip:** To refer to the directory containing the script (if any), use: `!{script:dir}`. (Note: An earlier version of this command, `!{program:dir}`, is also supported). This allows scripts to refer to data in the same directory, even if the directory is moved or if SeqNinja is launched from a different location.

### Examples:

Examples related to defining custom settings are shown in the table below.

When you create a script in the [Visual view](#) by adding “steps,” a corresponding piece of scripting language is added to the script. This underlying script can be seen by switching to the [Text view](#), where it can also be edited. You can then review the steps in the Visual view, if desired, before running the script.

For instance, if you added the script in the second row (“Sequence literals”) in the Text view and then looked at it in the Visual view, it would appear as:



Context	Example	Comment
<b>Paths</b>	"out.fap" = "!{program:dir}/adh4_mouse.pro"	
<b>Sequence literals</b>	"out.fasta" = "!{1}" +	In this example, the user passes a sequence of residues as the first command-line argument.



	<code>"in.fasta"</code>	<p>For literals defined in the script, it's easier just to use a sequence variable:</p> <pre>\$INSERTION = "NNNNNNNNN"</pre> <pre>"out.fasta" = \$INSERTION + "in.fasta"</pre>
<b>Patterns</b>	<code>"out.fasta" = collect('!{script:dir}/genomes/Vibrio*.gbk')</code>	
<b>Sequence index</b>	<code>"out.fasta" = "454.fasta"#{1}"</code>	
<b>Settings</b>	<pre>!mydir="C:\Users\norville"</pre> <pre>!mydata="!{mydir}/data"</pre> <pre>!myresults="!{mydir}/results"</pre>	
<b>Integers</b>	<pre>out.fasta = in.gb (</pre> <pre>  %!{1} &gt; %!{2} )</pre>	<p>This allows using integers passed on the command line in numeric expressions. For values defined in the script, it's easier just to use variable integers:</p> <pre>%A = 42</pre> <pre>%B = 111</pre> <pre>out.fasta = in.gb( %A &gt; %B )</pre>
<b>Single-quoted strings</b>	<code>print( 'File: !{1}' )</code>	

# Use editing commands



The following editing commands may be available, depending on the view, template or toolkit item being used.

Function	Menu command	Shortcut	Right-click shortcut in <a href="#">Visual view</a>	Notes
Remove a portion of text and place it on the clipboard.	Edit > Cut	Ctrl/ Cmd+X	Cut	
Copy a portion of text and place it on the clipboard.	Edit > Copy	Ctrl/ Cmd+C	Copy	
Paste an item from the clipboard at the cursor insert point.	Edit > Paste	Ctrl/ Cmd+V	Paste	
Delete a portion of text without placing it on the clipboard.	Edit > Delete	Delete key	Delete	
Select all text	Edit > Select All	Ctrl/ Cmd+A	Select All	
Undo the previous command	Edit > Undo	Ctrl/ Cmd+Z		This command is disabled in the <a href="#">Text view</a> .
Cancel use of the “Undo” command	Edit > Redo	Ctrl/ Cmd+Y		This command is disabled in the <a href="#">Text view</a> .

# Run a script

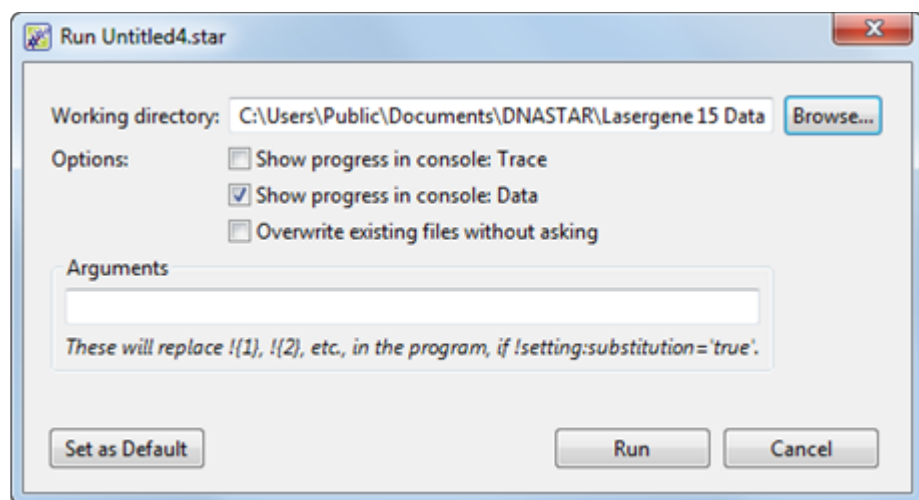
Once you have set up a script in the [Visual view](#) or [Text view](#) views, you can instruct SeqNinja to run the script from top to bottom using the Run command. The results will then be output into the data file and format that you chose in the script.

There are four ways to initiate a run. The first three initiate a run without allowing you to choose any options:

- Choose **Run > Run** from the menu.
- Click the **Run** tool (  ) in the toolbar at the top of either Script View.
- Click the **Run** tool (  ) below the script in the Visual View.

The fourth way to initiate a run allows you to choose options first:

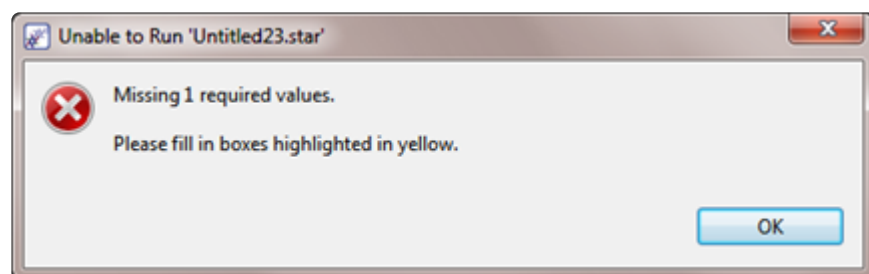
- Choose **Run > Run with Options** from the menu. This launches the following dialog:



- To change the working directory for both data and results, click **Browse** or type a new path and filename to the left of the Browse button.
- If, during a run, you want to view in the [Console](#) the step that is currently being processed, check **Show progress: Trace**.
- If, during a run, you want to view in the [Console](#) any activities related to data file use, check **Show progress: Data**.
- To turn off prompts warning that a file will be overwritten, check **Overwrite existing files without asking**. Note that the defaults for the first two options in this dialog can be set globally using the [Default Run Options](#) dialog.

- The Arguments text box can be used to provide arguments to the script. See [Use setting substitution](#) details and an example.
- To make the above set of choices be the default for this script in future runs, click **Set as Default**. If you do this, the chosen settings become the default regardless of which of the four methods you use to initiate a run in the future. To initiate the run, click the **Run** button. To return to the [Script view](#) without running, click **Cancel**.

Before initiating a run, all information must be complete. If you have neglected to fill in a box in the Visual view or if data is missing in the Text view, any attempts to initiate a run will generate a message like the one below:

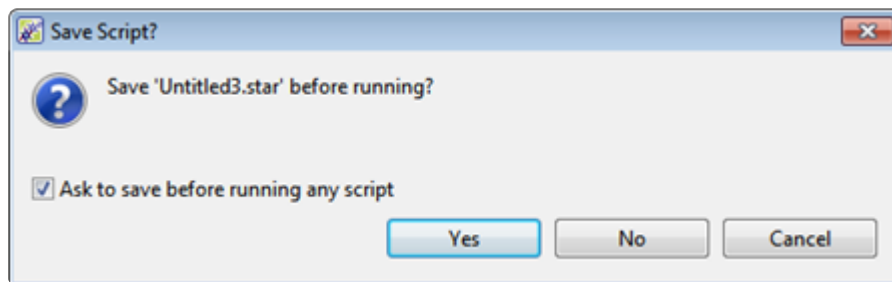


An error message will be generated in situations including:

- Input files that do not exist or that will not yet be created by the script before they're needed as input.
- Output files that are not writable.
- Undefined variables.
- Unknown functions.
- Missing or unexpected function arguments.

If you receive an “Unable to Run” message, click **OK**, and then address the issues before initiating the run again.

If you have not yet saved the script, or have made changes since the last save, you may be asked whether you wish to save the script.

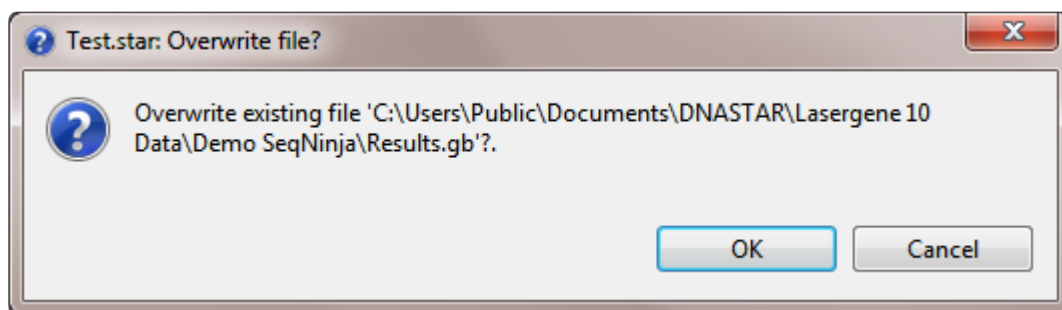


If the **Ask to save before running any script** is checked, this prompt will appear before running any script with unsaved changes. If you do not want to see this reminder in the future, uncheck this box.

✿ **Note:** You may also change your preference globally in the [Editor](#) dialog.

Click **Yes** to launch the Save dialog and save the script, **No** to run without saving, and **Cancel** to abort the run. If you respond affirmatively, the run will begin.

After the run begins, if you have chosen to send output to a filename that already exists, you will be asked if you wish to overwrite the existing file:




Click **OK** to overwrite and **Cancel** to abort the run.

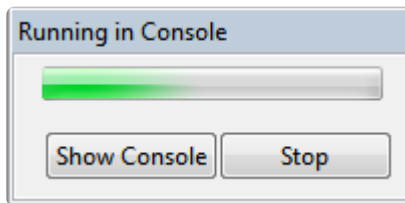
For information about what happens at the conclusion of a run, see [Access run results](#).

# Stop running a script

---

There are three ways to stop a run before it finishes:

- Choose **Run > Stop** from the menu.
- Click the **Stop** tool (  ) in the toolbar at the top of either Script View.
- By clicking the **Stop** button in this pop-up dialog.



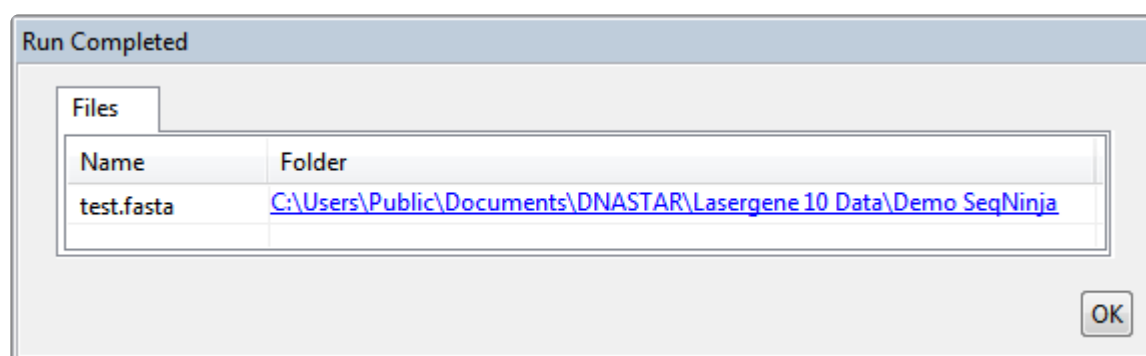
\* **Note:** SeqNinja executes many scripts so quickly that the “Stop” functionality never becomes active.

# Access run results

After [running the script](#), SeqNinja will display a Run Completed dialog. This dialog lets you navigate to files created by SeqNinja script, and also informs you of any issues encountered while running. The Run Completed dialog appears whether the run ends naturally or is terminated manually.

\* **Note:** As long as the [Script view](#) is still open at the conclusion of a run, the Run Completed dialog appears there. If the Script view has been manually closed, similar information instead appears in the [Console](#).

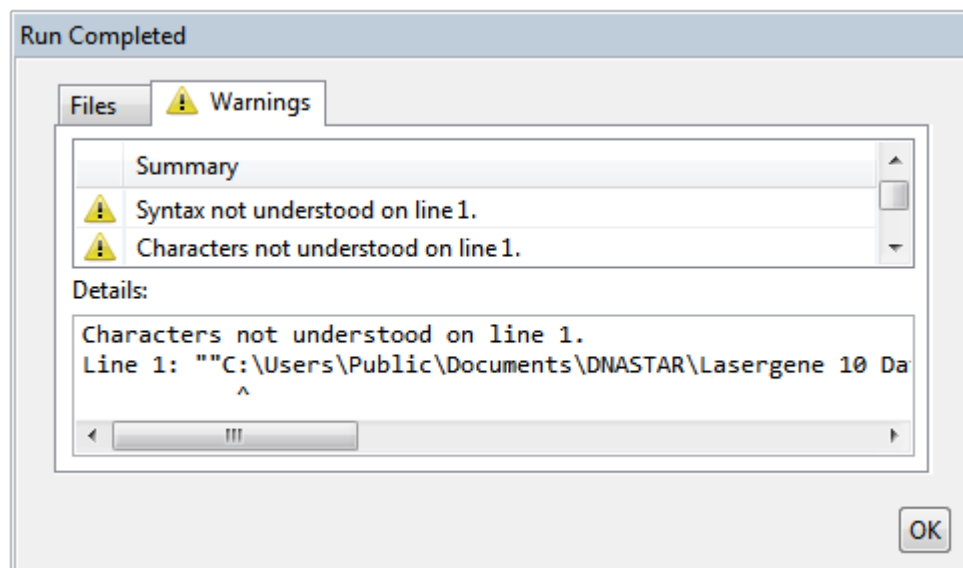
If the run finishes with no warnings, this dialog will have a single tab entitled **Files**:



A list of output files appears in the **Name** column, while their paths appear in the **Folder** column. Click the **Folder** column link(s) to navigate to the folder(s) with your saved results.

\* **Note:** If the number of output files exceeds 1000, only the directories to which they were written will be displayed.

If any warnings were generated during a run, the Run Completed dialog will have two tabs: **Files** and **Warnings**. Click the **Warnings** tab to view information about errors that prevented the run from completing successfully. The Summary section at the top contains a list of single-line warnings. Click any item in the Summary list to view complete information about it in the Details section. If the issue is related to syntax, the Details section may indicate the source of the error with a caret (^), as shown in the example below.



✿ **Note:** The [Console](#) may contain similar or additional information regarding warnings.

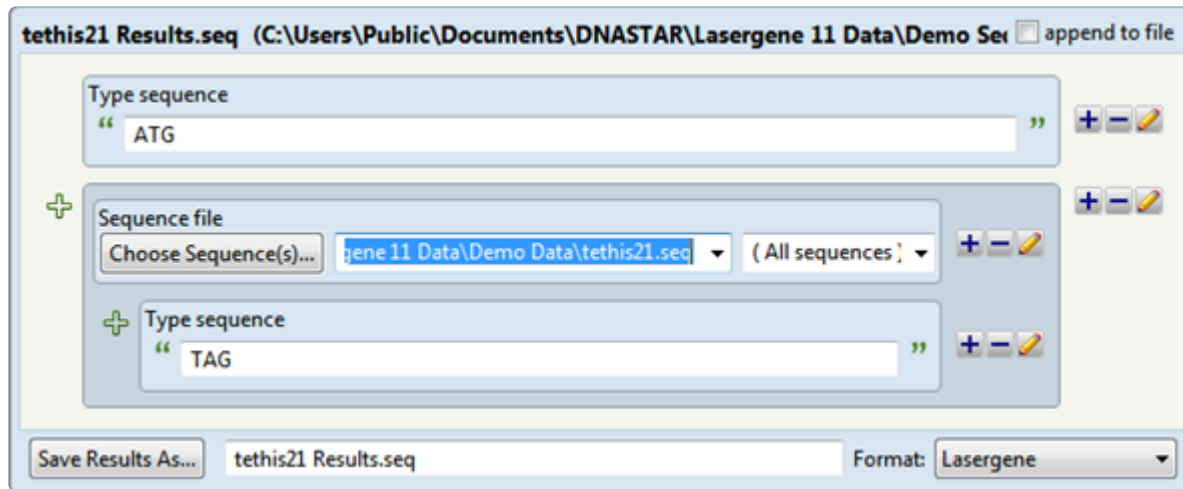
Click **OK** to close the message. After rectifying any errors, you can then initiate a new run.

Once the run has finished, you may open the results file in a text editor or another DNASTAR application. If the output format allowed it (GenBank files; Lasergene files; FASTA files when [Set Full File Conversion](#) = full), SeqNinja will automatically have updated the Locus section and added the following sequence history to the Comments section:

- The time of creation of the sequence.
- The name of the sequence (if applicable).
- The SeqNinja expression that can be used to create the sequence.

### Example:





The output file *tethis21 Results.seq* includes a sequence history showing how the sequence was created.

```

LOCUS      Out:1                911 bp    DNA                19-DEC-2012
COMMENT    Sequence generated by SeqNinja on Dec 19, 2012 at 5:07:54 PM CST.
           Equivalent SeqNinja expression:
           "ATG" + "C:\Users\Public\Documents\DNASTAR\Sample Data\Demo Data\In.seq"#"X05543" + "TAG"

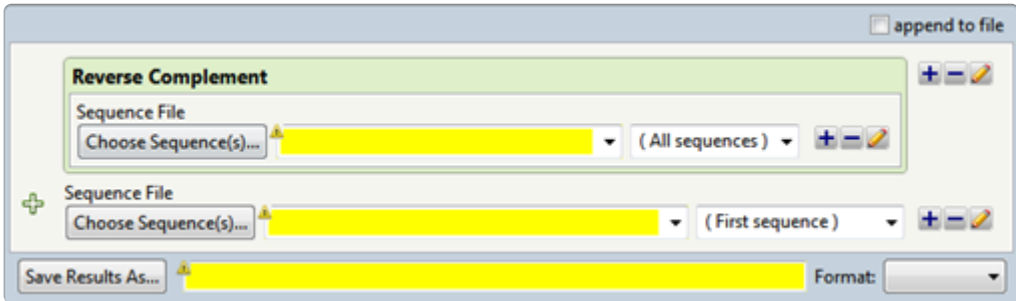
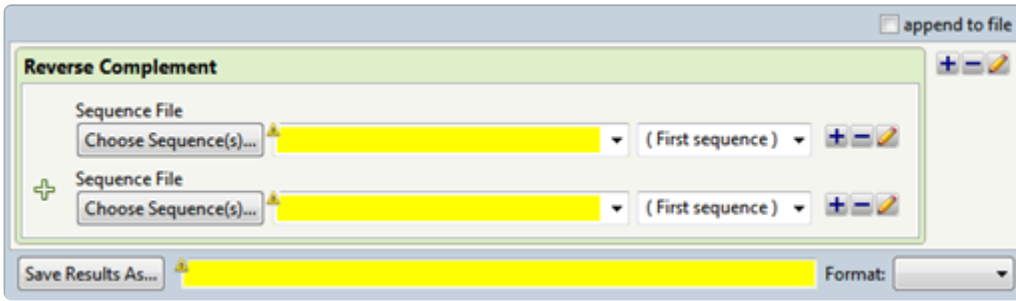
```

# Use drag & drop

Drag & drop functionality varies depending on which item you are dragging and where you wish to drop it.

- [Settings section](#) items, [Other section](#) items, or the two upper-most [Sequences section](#) items – Dragging & dropping these items on or above an existing script causes them to be placed above any existing steps. Dropping them at the bottom of the script positions them below any existing steps.
- The four lower-most [Sequences section](#) items (the “Add” operations) – Dragging & dropping these items on different parts of an existing script adds to a concatenation of sequences.

The following table shows the results of dragging and dropping the [Add Sequence File](#) tool on or near a [Reverse Complement Sequence](#) template:

Method for applying the Add Sequence File tool	Result
Single-click the <b>Add Sequence File</b> tool (drag & drop is not used)	Calculates the reverse complement of Sequence #1, then adds it to the original version of Sequence #2.
Drag and drop the tool below the template	
Drag and drop the tool when the green function area is highlighted.	Calculates the reverse complement of “Sequence #1 plus Sequence #2.”
Drag and drop the tool when the row with a <b>Choose Sequence(s)</b> button is highlighted.	

- [Functions section](#) items – Drag & drop can be used to nest these items inside almost any other step or operation in a script. An example of a nested function is shown below.

☐ append to file

**Translate**

Sequence file  
Choose Sequence(s)... (All sequence)

**Extract**

Features: Filter: =

Select sequence source

File Sequence Text Variable Function

Translation Table: <Default>

Start Codon: <Default>

Save Results As... Format:

# Save, Close and Exit

---

## Saving scripts:

- To save a new script in the SeqNinja *.star* format, or to overwrite a previously saved version of a script, choose **File > Save**, or use **Ctrl/Cmd+S**.
- To preserve the earlier version of a *.star* script or store a copy of the script with a different name or location, use **File > Save Results As**. By default, SeqNinja saves to the desktop unless directed to save elsewhere.
- To simultaneously save all open SeqNinja scripts as *.star* files, use **File > Save All** or press **Ctrl/Cmd+Shift+S**.

## Closing the active project:

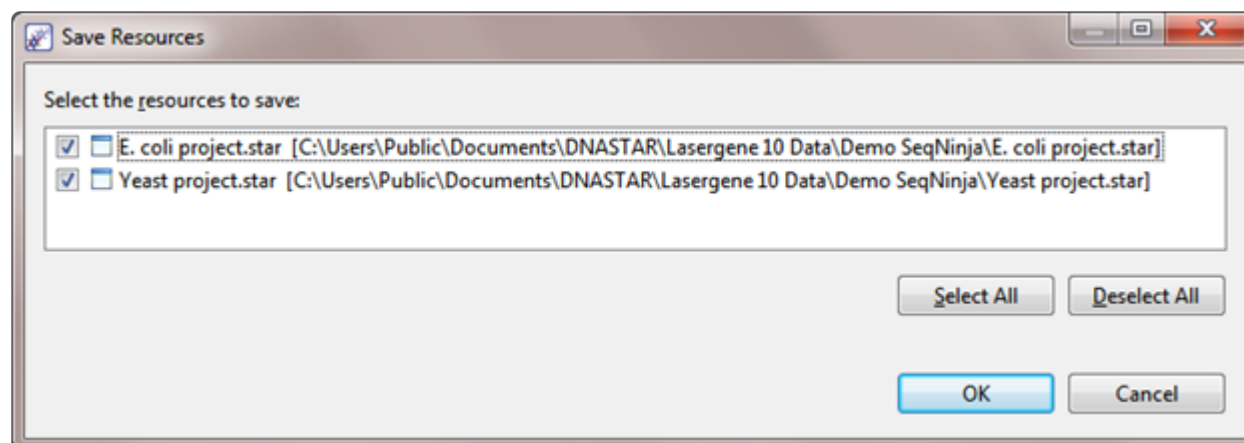
Use **File > Close** or press **Ctrl/Cmd+W**.

## Exiting the SeqNinja application:

On Windows, use **File > Exit**. On Macintosh, use **SeqNinja > Quit SeqNinja** or press **Cmd+Q**.

If only one script is open when you exit, SeqNinja will open the Save Document dialog.

If there are multiple edited and unsaved scripts open when you exit, SeqNinja will instead display the Save Resources dialog.



- If you wish to save all changed scripts, choose **Select All**. If you wish to exit from all scripts without saving them, select **Deselect All**. To save only some of the scripts, add/remove checkmarks individually.

- Click **OK** to exit or **Cancel** to return to SeqNinja without saving any scripts.

# Miscellaneous

---

# Set preferences

---

SeqNinja lets you set preferences using a multi-tabbed dialog. To access the dialog on Windows, use **Edit > Preferences**. On Macintosh, use **SeqNinja > Preferences** or press **Cmd+,**.

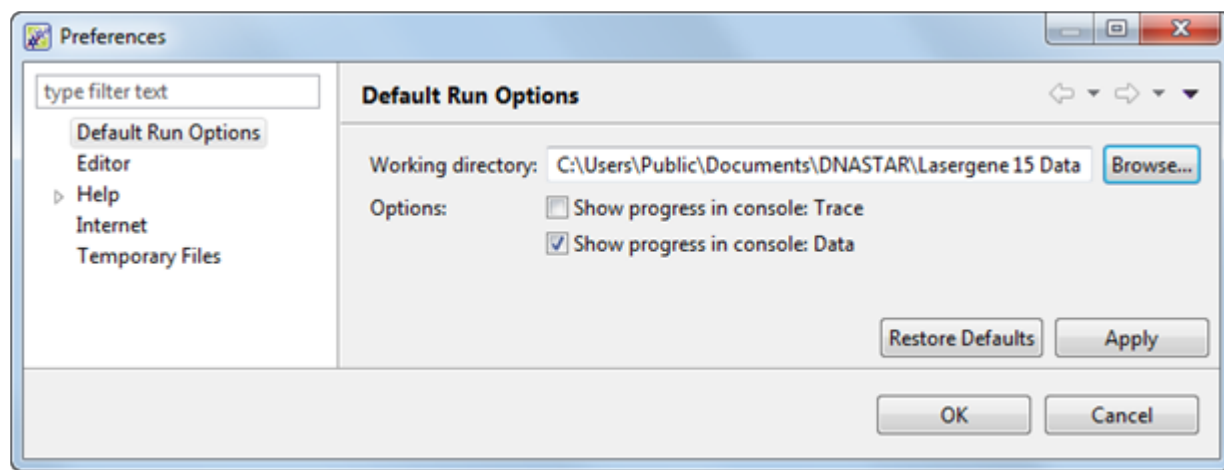
To learn about each tab of the dialog, see:

- [Default Run Options](#)
- [Editor](#)
- [Help](#)
- [Internet](#)
- [Temporary Files](#)

# Default Run Options

The Default Run Options dialog allows you to choose a default results folder, and to elect whether you wish to show two facets of a run's progress in the [Console](#).

The dialog is accessed via the command **Edit > Preferences** (Win) or **SeqNinja > Preferences** (Mac). If the Default Run Options section is not already active, click on **Default Run Options** on the left.



- Use the **Browse** button to the working directory for data and results. This working directory will be used during script execution as the default location for a `!datapath*` specified with a relative path, and also as the default location for files specified with a relative path, if no appropriate `!datapath*` is set. Note the setting this preference for a working directory does not affect the directory shown by file choosers during the editing of scripts. File choosers instead show the last directory from which a file was chosen.
- If, during runs, you want to view in the [Console](#) the step that is currently being processed, check **Show progress: Trace**.
- If, during runs, you want to view in the [Console](#) any activities related to data file use, check **Show progress: Data**.

Changes in this dialog will apply when running future scripts. Scripts that have already been run at least once will continue using their original run options.

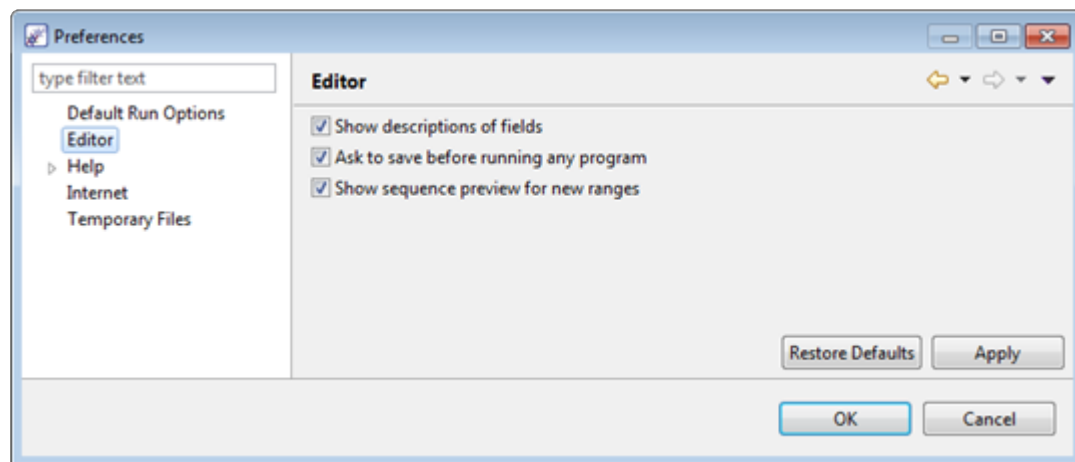
Click **Apply** and **OK** to save your changes or **Cancel** to ignore any changes. Click **Restore Defaults** if you would like to return to the default settings.



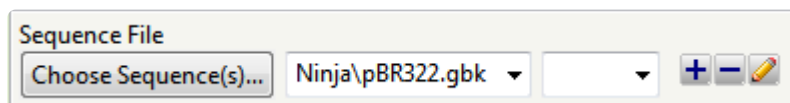
# Editor


The Editor Preferences dialog controls whether or not you are prompted to save changes prior to running a script, and whether or not “field descriptions” are shown by default.

The dialog is accessed via the command **Edit > Preferences** (Win) or **SeqNinja > Preferences** (Mac). If the Editor section is not already active, click on **Editor** on the left.

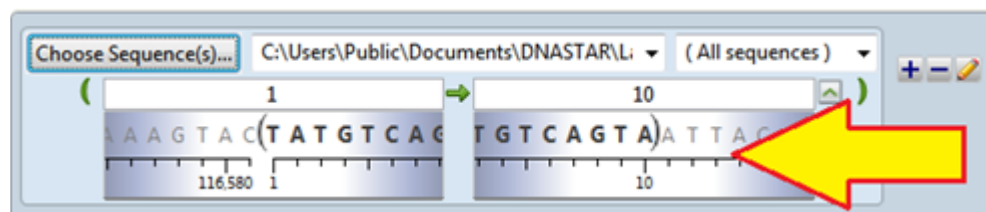


- Check or uncheck **Show descriptions of fields** depending whether you would like to show or hide field descriptions by default. Field descriptions are labels such as the words “Sequence File” in the image below:



Note that you can over-ride this default at any time using the **Show/Hide descriptions of fields** tool (  ) above the [Script views](#).

- Check **Ask to save before running any script** if you wish to receive a “save” prompt before running any script with unsaved changes. If you do not want to see this reminder in the future, uncheck this box.
- Check **Show sequence preview for new ranges** to automatically display the range wheels (see image) for a sub-range step.



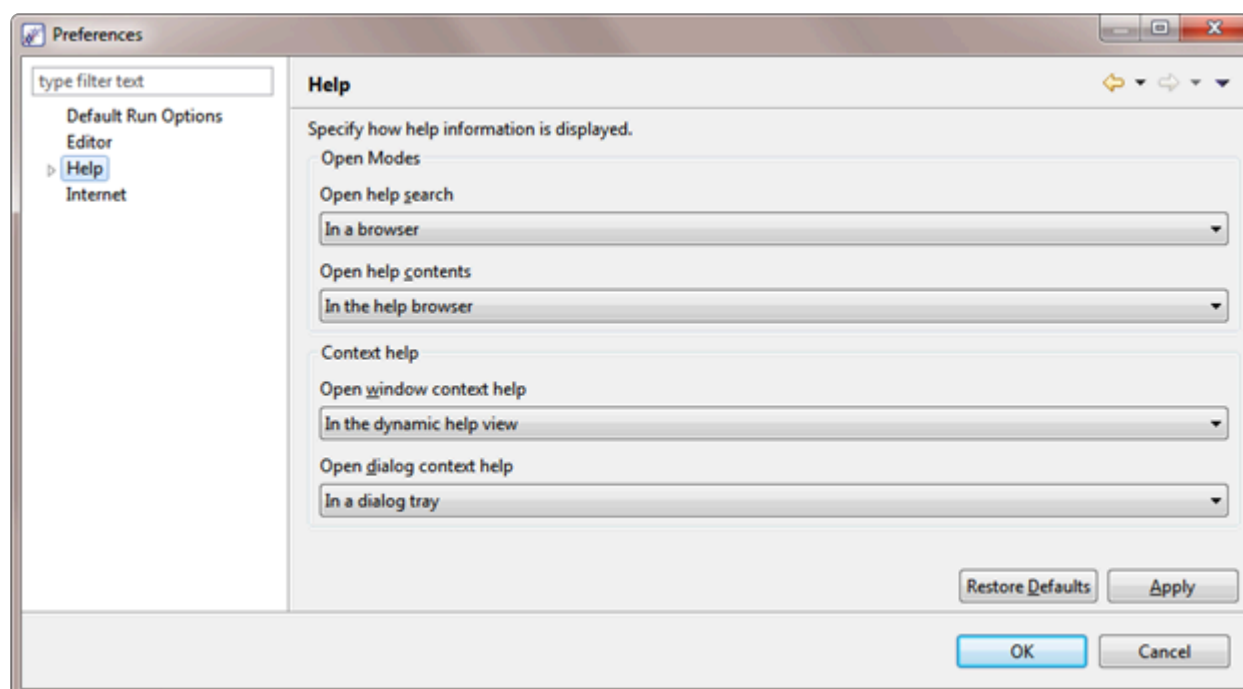
If the box is left unchecked, the range wheels can be displayed manually. See [Set ends for a sequence](#) for instructions.

Click **Apply** and **OK** to save your changes or **Cancel** to ignore any changes. Click **Restore Defaults** if you would like to return to the default settings.

# Help

The Help Preferences dialog controls how help documentation is displayed when you use the **Help > Help Contents** command.

The dialog is accessed via the command **Edit > Preferences** (Win) or **SeqNinja > Preferences** (Mac). If the Help section is not already active, click **Help** on the left. Note that there is no need to use the expand arrow to the left of “Help.” Its only sub-menu item, “Contents,” is disabled in SeqNinja.



- Specify the browser used to display the SeqNinja Help search results by selecting one of the following options from the **Open help search menu**:
  - **In the dynamic help view** to display the SeqNinja Help search results in a new section of the Help interface. This is the default setting.
  - **In a browser** to display the SeqNinja Help search results in your default web browser.
- (On Windows only) Choose how to display the SeqNinja Help contents by selecting one of the following options from the **Open help contents menu**:
  - **In the help browser** to display the SeqNinja Help content in a new section of the Help interface. This is the default setting.
  - **In an external browser** to display the SeqNinja Help content in your default web browser.

The ContextHelp options in the lower half of the dialog are not applicable to SeqNinja.

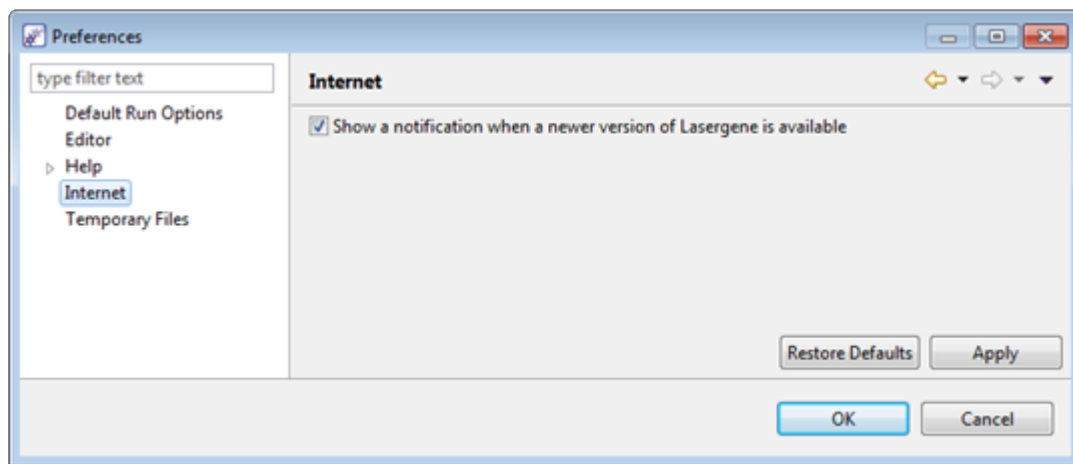
Click **Apply** and **OK** to save your changes or **Cancel** to ignore any changes. Click **Restore Defaults** to return to the default settings for all four drop-down menus.

# Internet

---

The Internet Preferences dialog lets you choose whether you wish to be notified about Lasergene updates.

The dialog is accessed via the command **Edit > Preferences** (Win) or **SeqNinja > Preferences** (Mac). If the Internet section is not already active, click on **Internet** on the left.



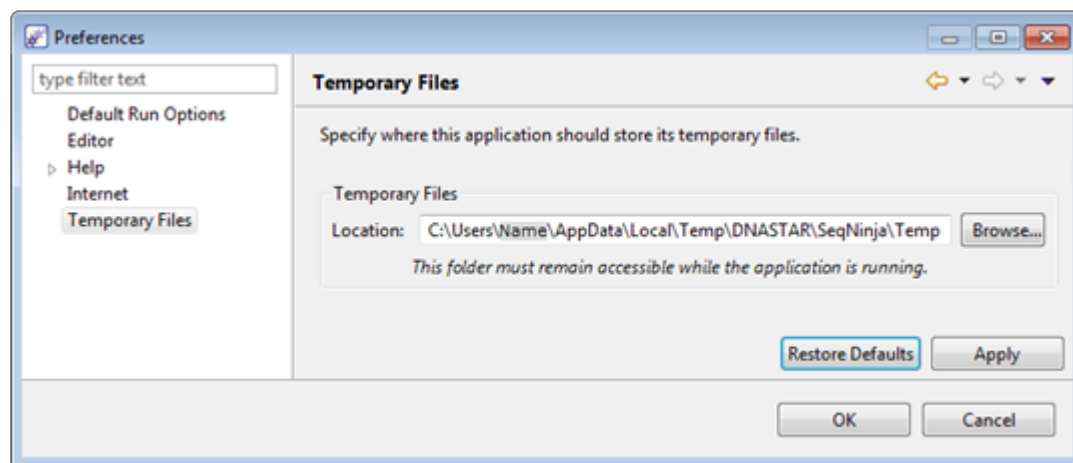
Check the box if you would like SeqNinja to display a notification when a newer version of Lasergene is available. Uncheck the box if you do not wish to receive this notification.

Click **Apply** and **OK** to save your changes or **Cancel** to ignore any changes. Click **Restore Defaults** to return to the default settings for all four drop-down menus.

# Temporary Files

SeqNinja creates and uses temporary files while a project is open. The need for temporary file space grows with the number and length of the sequences being manipulated. If you expect the temporary files will be too large to fit in the default location, you may specify a different location for these files using the Temporary Files Preferences dialog.

The dialog is accessed via the command **Edit > Preferences** (Win) or **SeqNinja > Preferences** (Mac). If the Temporary Files section is not already active, click on **Temporary Files** on the left.



To change the temporary file location, close all SeqNinja projects and then return to this dialog. Select the **Browse** button, navigate to the new location and click **OK**. As noted in the dialog, the location you choose must remain accessible while SeqNinja is running.

Changes in this dialog will apply when running future scripts. Existing scripts that have already been run at least once will continue using their original temporary file location.

Click **Apply** and **OK** to save your changes or **Cancel** to ignore any changes. Click **Restore Defaults** if you would like to return to the default settings.

# Turn off usage logging

---

By default, [usage logging](#) is enabled in Lasergene version 11 and later. To opt out of usage logging, launch the DNASTAR Navigator and go to **View > Preferences > Lasergene** (Win) or **DNASTAR Navigator > Preferences > Lasergene** (Mac). Uncheck the box and click **Apply** and then **OK**.

# Clean cached files on startup

---

Prior to the Lasergene 14.0 release, the default SeqNinja startup ran a `-clean` option. This option automatically removed some cached files on application restart.

The `-clean` option is no longer executed at each startup. However, if you are a Windows user, you may execute an explicit `-clean` in the unusual event that cached data has become corrupted:

1. Close SeqNinja if it is currently open.
2. Launch the Command Line (*C:\Windows\system32\cmd.exe*).
3. Type the path to the SeqNinja application, followed by a space and the `-clean` command.

```
C:\Program Files (x86)\DNASTAR\DNASTAR\Lasergene 14\SeqNinja.exe -clean
```



# SeqNinja Command-Line Help

---

# Getting Started with SeqNinja Command-Line

The command-line version of SeqNinja runs within a command shell (terminal) window. SeqNinja is launched in different ways, depending whether you'd like to run in "batch mode" or "interactive mode." All scripting language commands can be used in either mode.

- In batch mode, you run a command script stored in a file. Batch mode is useful for complex scripts and for repeatability. The script file provides a record of the operations performed. An example of a script for batch mode use is shown below:

```
concatenate_contigs.star
; Add stop codons in all frames between contigs
!datapath="C:\Users\Public\Documents\Data"
$L = "NNNNNCACACACTTAATTAATTAAGTGTGTGNNNNN"
pseudo.fasta = contig00001.fasta
    +$L
    + contig00002.fasta
print('pseudomolecule: ', length(pseudo.fasta), ' bp')
```

- In interactive mode, you enter and run commands one-at-a-time, directly in the SeqNinja command shell. Interactive mode is useful for experimentation and brief, ad hoc commands. A sample interactive section is illustrated below.

Interactive  
SeqNinja

```
C:\>seqninja
SeqNinja 10.0.0.104
Copyright (c) 2011-2012 DNASTAR, Inc.
Subscriber Number: 000000

Type 'help' for more information.

:: help
:: !datapath="C:\Users\Public\Documents\Data"
:: $L = "NNNNNCACACACTTAATTAATTAAGTGTGTGNNNNN"
:: pseudo.fasta = ctg1.fasta + $L + ctg2.fasta
:: print('Length=', length(pseudo.fasta), ' bp')
Length=213770 bp
:: exit
Exiting SeqNinja shell.

C:\>
```

#### Running SeqNinja in batch mode:

From within a Command Prompt window (Win) or Terminal window (Mac), type “seqninja” followed by the name of a script, or by another [command-line option](#).

If running a script, you will first need to prepare a file of scripting commands using the SeqNinja application or any text editor, and then save the file with the extension .star (e.g., *myfile.star*). In the command shell, enter the command `seqninja filename.star` and press **Enter**. This will run all the scripting commands in order of their appearance in the text file.

#### Running SeqNinja in interactive mode:

Launch the SeqNinja command shell in either of these ways:

- From within a Command Prompt window (Win) or Terminal window (Mac), type `seqninja`.
- (Win only) From within a file explorer, double click on the SeqNinja executable (located in the SeqNinjaCL subdirectory).

Once the SeqNinja shell has been launched, you will see a double colon (i.e. : :) command prompt. You are now ready to begin entering scripting commands and pressing Enter to perform each operation.

✿ **Note 1:** If you launch the SeqNinja command shell from outside a command shell (terminal)

window, a command shell window will appear automatically. When you quit that SeqNinja session, the command shell window will disappear.



**Note 2:** The SeqNinja command shell does not support history or editing, other than backspace and “paste.”

#### Notification of scripting errors:

If there are issues with a script, SeqNinja will attempt to flag these prior to executing the script. SeqNinja checks the script for issues such as:

- Input files that do not exist, are not readable by the user, or are not created by the script before they're used as input.
- Output files that are not writable by the user.
- Undefined variables.
- Unknown functions.
- Missing or unexpected function arguments.

# Command-Line Options

The options below function from within the command shell, rather than from within the SeqNinja shell.

Objective	Expression	Comments
<b>Enter the SeqNinja shell or to execute a batch mode command</b>	<code>seqninja</code>	When used alone, launches the SeqNinja shell, allowing you to begin an interactive session. When followed by the name of a file or with one of the commands below, you will stay in the command shell and SeqNinja will execute the command or script in batch mode.
<b>Display a usage description</b>	<code>-h</code> or <code>-help</code>	Displays a short usage description. Using this command does not cause you to enter the SeqNinja command shell.
<b>Check the version number</b>	<code>-v</code> or <code>-version</code>	Displays the installed version number. Using this command does not cause you to enter the SeqNinja command shell.



**Note:** Using these expressions from within the SeqNinja shell may generate an error message.

# The SeqNinja Language

---

The SeqNinja language supports comments and statements. Statements include actions, assignments and settings.

- [Actions](#) are viewed as commands.
- [Assignments](#) store data in a file or variable.
- [Settings](#) configure options used implicitly throughout the rest of the script.

The following list outlines general conventions of the SeqNinja scripting language:

- In a script file, a statement may be spread over multiple lines, but may not be placed on the same line as another statement.
- Variables (e.g., file paths, etc.) are only stored in SeqNinja's memory until the end of the session, or until overwritten, whichever occurs first.
- White space and comments are always ignored.
- Single-line comments can be added after typing the characters `;` or `//`.
- Multi-line comments can be added after typing the characters `/*` or `* /`.
- SeqNinja supports non-ASCII characters in filenames, but non-ASCII characters may not be used as variables.
- Scripts with non-ASCII characters, including international characters, must be saved with the UTF-8 encoding, without byte order marks.
- Double quotes should be used around individual file paths, whether in settings or sequence expressions. Double quotes *inside* of paths are never supported.
- Single-quotes should be used to signify a file pattern.
- Forward slashes are supported in file paths on both Windows and Macintosh. Backslashes are supported only on Windows. If a script is intended to be run on both systems, use forward slashes in file paths.

# Escape codes

All quoted strings accept certain “escape codes.” These escape codes indicate that SeqNinja should put a tab, a new line, a quotation mark, etc. in the specified location:

Escape Code	Effect
<code>^t</code>	Inserts a tab.
<code>^n</code>	Inserts a line break.
<code>^^</code>	Inserts a caret character (^).
<code>^'</code>	Inserts a single quote into a string that is bracketed by single quotes.
<code>^"</code>	Inserts a double quote into a string that is bracketed by double quotes.

Example input:

Using print with the following argument causes the name of the file, length of the sequence, and the sequence itself to print in the output stream in a neatly organized manner.

```
print( 'sequence file name:^t', 'contig01.fas^n', 'sequence length:^t', length("contig01.fas"), '^n', 'sequence', '^n', "contig01.fas" )
```

Example output:

```
sequence file name:   contig01.fas
sequence length:     116581
sequence
TAATGTCAGTAATTACCGCGTTTCGCCAGCGTCAGTTCTCTGGCATTITTTGTCGCGCTGGGC
TTTGTAGGTAATGGCGTTATCACGGTAATGATTAAACAGCCCATGACAGGCAGACGATGAT
GCAGATAACCGAGAGCGGAGATAATCGCGGTTACTCTGTTTCATTGCTGACCCCAACAG
ATTTACGCTCAATCTACGACGAGTCATGAGACCTTTCCATTGCTTACCGCCAGCATAT
GTCCAGCGACGTAGCTGATCACATGCGCCTTTGATATCGCCCTGGTTTATTTTGCAGAGA
```

# File patterns

---

A file pattern is a search string with one or more wildcards in the filename. The [collect sequence function](#) and [split action](#) both support file patterns as arguments.

The following are some rules pertaining to the use of file patterns:

- A question mark (?) wildcard matches exactly one arbitrary character. An asterisk (\*) wildcard matches zero or more arbitrary characters. Split only allows asterisk (\*) wildcards. Collect allows both asterisk (\*) and question mark (?) wildcards.
- A file pattern may be relative or absolute. Relative paths in patterns are resolved in the same manner as for double quoted path names.
- Wildcards can only be used in the filename, and nowhere else in the path.
- Single quotes must be used to distinguish file patterns from individual file paths, which are double quoted. This makes it possible to reference files whose names contain wildcard characters.



# Settings


Settings configure values used implicitly throughout the rest of the script, and always begin with an exclamation mark (!). The value of settings may be a single word, or a quoted string using single or double quotes. In a setting, double quoted strings are not interpreted as sequences.

Built-in settings are shown in the table below. You can also [use setting substitution](#).

Objective	Expression	Examples	Comments
Set the root directory for relative file paths used in the rest of the script	!datapath !datapath:in !datapath:out	!datapath='C:\Users\MyName\data\'	<p>Relative path names are presumed relative to one of the following, in order:</p> <ul style="list-style-type: none"> <li>• If set, !datapath:in (for input files) or !datapath:out (for output files).</li> <li>• Otherwise, if set, !datapath.</li> <li>• Otherwise, the working directory.</li> </ul> <p>Using different directories for input and output reduces risk of accidentally overwriting an input file.</p>
Substitute setting values inside strings	!setting	!setting:substitution=true	'True' permits the substitution of setting values in a script; 'false' prohibits such substitution.
Specify a temporary files directory	!tempdir	!tempdir='D:/temp'	This can be used to put temporary files on a drive with more space, or in a more accessible location.
Filter output features to those listed	!features	!features='CDS,source'	See the notes below this table.

Set verbose output of the types specified in the value	<code>!verbose</code>	<code>!verbose='TRACE,DATA'</code>	This sets verbose output of the types specified in the value. This can be useful for providing progress information and for troubleshooting.
--	-----------------------	------------------------------------	--

Objective	Expression	Examples	Comments
Output an auxiliary file to preserve features during format conversion	<code>!conversion</code>	<code>!conversion=natural</code> <code>!conversion=full</code>	See <a href="#">Convert Between Annotated and Featureless Formats</a> . Also see the notes below this table.
Specify whether to record a sequence history in the comments section of generated sequences	<code>!comments:history</code>	<code>!comments:history='on'</code> <code>!comments:history='off'</code>	<p>Sequence history is only written when comments can be written to the output format. A sequence history can be written for GenBank files, Lasergene files, and FASTA files (when <code>!conversion='full'</code> is set).</p> <p>A sequence history includes:</p> <ul style="list-style-type: none"> <li>• When the sequence was created.</li> <li>• The name of the sequence, if it has a name before written.</li> <li>• A SeqNinja expression that can be used to create the sequence.</li> </ul>

 **Note 1:** Other settings can affect the display of features resulting from use of `!conversion` and `!features`. For example, including `!features=CDS` above a statement with range endpoints will cause only CDS features to be displayed.

✿ **Note 2:** If you wish to set endpoints based on features that you want to later filter from the output, temporarily set `!features=*`. After setting the endpoints, change the `!features` setting to the type required for the desired output.

# Integral expressions

Supported integral operations are shown below, and always begin with a percent sign (%).

Objective	Operators	Examples	Comment
Add, multiply and divide	+ * /	<code>%i=1+2*4</code>	Operands and results are always integers, and any fraction obtained through division is discarded.
Make an assignment	=	<code>%i=1+2</code>	This overwrites any earlier assignment.
Assign a variable to an integral expression	<code>%variable_name</code>	<code>%i=3+4+%j</code>	Legal characters in a variable name are A-Z, a-z, 0-9, and _ (underscore).
Measure length	length	<code>%len=length h("H:\MG1655.fasta")  print(length h(\$M+\$A))</code>	If the length argument contains multiple sequences, only the first is used.
Count sequences	countSequences	<code>%s=countSequences ("H:\454.fasta")</code>	Total count across the entire set of sequences provided.

# Sequence expressions and assignments

---

A sequence expression describes a sequence or a set of sequences. Sequence expressions include:

- A [sequence filename, a literal sequence, or a variable](#) representing the value of a sequence expression.
- A reference to a single sequence in a [multi-sequence file](#) by path/name, name or number.
- A [fragment](#) (subrange) of a sequence, where each coordinate is specified by a “site expression” (also see [Integral Expressions](#)).
- A [concatenation](#) of sequence expressions.
- A [function](#) of a sequence expression, such as complement, cut, extract or translate.

# Sequence filenames, literal sequences and variables

One or more sequences can be obtained from a filename or variable or provided as a literal sequence. Sequence variables always start with a dollar sign (\$).

Objective	Expression	Examples	Comments
Write any sequence expression to a sequence file	<code>filename=sequence_expression</code>	<code>myfile.fasta=\$b</code>	Causes output to be written to the specified file, in a format determined by the specified file's extension. If you use the <code>=</code> expression and the filename already exists, you will be asked whether you wish to overwrite it. Appending is supported via the <code>+=</code> expression (see next row).
Append data to an existing	<code>filename+=sequence_expression</code>	<code>myfile.fasta+= \$b</code>	This adds data to an existing file

sequence file			rather than overwriting the existing data.
Convert to another format	sequence_setA=sequence_setB	alpha.fasta=alpha.gb	
Convert a set into a data file	sequence_setA=sequence_setB.star	my.fasta=my.star	
Assign a sequence expression to a variable	Simple filenames	\$a=myfile.fasta	Double quotes are necessary for filenames that include characters other than A-Z, a-z, 0-9, the underscore, or '.' after the first character.
	Filename with spaces	\$b="my filename.fasta"	
	Filename with slashes	\$c="C:\data\MG1655-e-coli-k-12substrands.fasta"	
	Unicode filename	\$d="Мое имя файла.fasta"	
	Filename with diacritic mark	\$e="ma séquence.fasta"	
Assign a variable or filename to a literal sequence	"sequence"	\$f="ACGT"	Where a sequence expression might occur, double quoted string literals are interpreted as either sequence literals or sequence filenames, depending on whether they
	sequence_set="sequence"	alpha.fasta="ACGT"	

			contain characters outside A-Z and a-z.
--	--	--	--

✿ **Note 1:** The output file format depends on the filename extension on the left-hand-side of the assignment. A recognized filename extension is required.

✿ **Note 2:** Within the same script, you may use the output sequence from an earlier step as the input sequence for a later step. However, the result might not be what you expected or desired. For example, if an extracted CDS initially had a /codon\_start annotation (i.e., it did not start with an intact codon), that information would be lost in the extracted FASTA file. Then, when the file was subsequently translated, it would be in the wrong reading frame.



# Multi-sequence files and expressions

SeqNinja can read and write files that contain multiple sequences. For example, a sequence expression might be the name of a file containing multiple sequences. Some operations support multiple-sequences, while others operate only on the first. For example, reverse-complement supports multiple sequences. But concatenation works only on the first sequences of its operands.

To specify a single sequence from a multiple-sequence file, use one of these forms:

Objective	Expression	Examples	Comments
Pick a sequence by numerical index	filename#index	"WISYPE_1.fasta"#2 "WISYPE_1.fasta"#2(1,20)	The first sequence in the file is index 1.
Pick a sequence by name	filename#name	"WISYPE_1.fasta"#"Chromosome_contig2"	This form works best if all the sequences in the set have unique names. If multiple sequences have the same name, then this expression will match the first encountered sequence whose name matches.
	filename#"name"	"WISYPE_1.fasta"#"Chromosome_contig2"(1,20)	
	filename/sequenceName	"WISYPE_1.fasta"/"Chromosome_contig2"	This refers to the sequence in the file as if it were a file in a directory.

✿ **Note 1:** Some special characters in sequence names may require use of the # notation.

✿ **Note 2:** Accessing multiple individual sequences by index from a multi-sequence file can be faster when the access is done in the order those sequences appear in the file. For example, if a file contains sequences (a,b,c,d,e,f), then accessing (a,c,f) may be faster than (f,c,a).

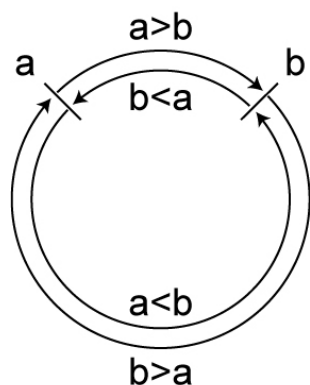
# Sequence fragments

Following some sequence expressions, a fragment (sub-range) can be specified using coordinates. Here are some rules pertaining to sequence fragments:

- Each coordinate is specified by a “site expression.”
- Integral coordinates are supported. The first base has index 1.
- The full expression is always evaluated left-to-right.
- A sub-range applies to all sequences in a multi-file sequence. Ranges are re-evaluated for each member of the set to which they are applied.

Objective	Expression	Examples
Read right from coordinates a to b, inclusive	(a > b) (a,b)	\$B=alpha.fast a(1,414) \$F=alpha.fast a(%i+3, 414) \$C1=\$A(1,41 4)=\$A(1 > 414)
Circle right	(b > a)	\$E1=\$A(414 > 1)
Circle the reverse complement	(a<b)	\$E2=\$A(1<414)
Read the reverse complement left from coordinates b to a	(b<a)	\$D1=\$A(414<1)
Specify the left end and right end bases	lend rend	\$D=\$A(lend,rend) \$D=\$A(lend+1, rend+1)
Reference a base position relative to the left end of the sequence. (Note that “left” can only be used to specify the coordinate on the right-hand side.)	left	out.fasta = in.fasta(lend, lend + 12)

See the image below for a visual representation of the different sequence segments.



Coordinates can also be specified by the results of one or more searches for matching sequence. The value of a search is a coordinate. Site expressions can include both searches and integral expressions, all evaluated left-to-right.

IUPAC ambiguity codes are recognized both in the sequence and in the expression. For example, typing AAS (where S = C or G) into an expression will cause SeqNinja to look for the first instance of AAC or AAG in the sequence. Conversely, typing AAC into an expression will cause SeqNinja to look for the first instance of AAC, as well as any combination of bases and ambiguity codes that would allow for AAC (e.g., AAS, WWM, etc.).

In the following search examples, \$A="AAAGGGCATCCCTTT":

Objective	Expression	Examples	Result
<b>For left position, start at lend, search forward</b>	(lend > > "sequence", rend)	\$A(lend > > "CAT", rend)	CATCCCTTT
<b>For right position, start at lend, search forward</b>	(lend, lend > > "sequence")	\$A(lend, lend > > "CAT")	AAAGGGCAT
<b>For left position, start at rend, search reverse complement</b>	(rend<<"sequence", rend)	\$A(rend<<"ATG", rend)	CATCCCTTT
<b>For right position, start at rend, search reverse complement</b>	(lend, rend<<"sequence")	\$A(lend, rend<<"ATG")	AAAGGGCAT
<b>Search forward implicitly from lend</b>	("sequence", rend)	\$A("CAT", rend)	CATCCCTTT
<b>Search forward implicitly from left position in direction/strand of single arrow</b>	(lend+number > "sequence")	\$A(lend+3 > "CAT")	GGGCAT
<b>Search forward implicitly from right position in direction/strand of single arrow</b>	(rend<"sequence")	\$A(rend<"ATG")	AAAGGGATG
<b>Look forward for first "CC", then look forward from there to next "T" (chained</b>	(lend > > "seq	\$A(lend > > "C	TTT

<b>search)</b>	<code>a" &gt; &gt; "seq b", r end)</code>	<code>C" &gt; &gt; "T", ren d)</code>	
<b>Find ambiguity code matches</b>	<code>("sequence", ren d)</code>	<code>\$A("CAN", rend) \$A("CAW", rend)</code>	<b>CATCCCTTT</b>
<b>Search for “CAT” and add 3; this mixes search operators with arithmetic</b>	<code>("sequence"+numbe r, rend)</code>	<code>\$A("CAT"+3, ren d)</code>	<b>CCCTTT</b>

# Sequence concatenation

Sequence expressions can be concatenated with the + operator. Concatenation links together corresponding pairs in the left and right sequence sets. Following are some rules concerning sequence concatenation:

- When one of the sets is a singleton (for example, a literal), concatenation results in prepending or appending that single sequence to every sequence in the other set.
- If a concatenation includes more than one fragment from the same source sequence, then feature reassembly occurs when writing to a file format supporting features. Feature reassembly takes parts of the same feature in multiple fragments and combines them back into a single feature. When features are cut, reassembled, or (in some cases) offset, SeqNinja flags this with an annotation of the form: */note="\*\*\*needs review\*\*\*Explanation*, where *Explanation* is a short description of a potential issue (e.g., that the head or tail of a feature has been cut by a number of residues).
- A sub-sequence can cut a feature. Multiple sub-sequences concatenated together can temporarily result in multiple fragments of the same feature. SeqNinja can reassemble these fragments so that the output sequence has a single feature, and such fragments are reassembled if they do not overlap. Reassembly is performed left-to-right in the resulting sequence. For example, the “source” annotation in GenBank files can be cut into fragments by a SeqNinja expression. Before writing an output file, SeqNinja reassembles these fragments, as long as they don’t overlap. If a concatenation does not include multiple fragments from the same source, feature reassembly does not occur.

Expression	Example	Comment
+	<code>beta.fasta="ATG"+alpha.fasta+"CTA"</code>	Adds a prefix and suffix to another sequence.
	<code>gamma.fasta=alpha.fasta(31,41)+(59,265)+(358,979)</code>	Concatenates fragments from the same source without repeating the source name.

# Sequence functions

Functions resulting in sequence expressions always start with a dollar sign (\$) and include:

Objective	Expression	Example	Comments
Reverse-complement sequences, or to assign a file to the reverse-complement of another file	<pre>~(sequence_set)  complement(sequence_set)</pre>	<pre>\$rc=~(foo.fasta)  foo_rc.fasta=complement(foo.fasta)  MG1655_rc.fasta=~("C:\data\MG1655_e_coli_k_12substrands.fasta")</pre>	Though the command is “complement” (for brevity), SeqNinja is actually calculating the reverse complement of the selection.
Cut a sequence into smaller pieces	<pre>cut(sequence_set, size)</pre>	<pre>bar.fasta=cut(foo.gb, %i)</pre>	
Specify an overlap when cutting a sequence	<pre>cut(sequence_set, size, offset)</pre>	<pre>bar.fasta=cut(foo.gb, 180, 60)</pre>	This can be used to create faux reads from an assembled sequence.
Extract sub-sequences from sequences corresponding to the given features	<pre>extract(sequence_set, 'feature_type[,... ]')</pre>	<pre>bar.fasta=extract(foo.gb, 'CDS')  bar.fasta=extract(foo.gb, 'CDS, gene')</pre>	Single quotes are required for arguments other than the sequence set.
Extract matching features	<pre>extract(sequence_set, 'feature_type:/tag="value"[,... ]')</pre>	<pre>bar.fasta=extract(foo.gb, 'CDS:/gene="thrC"')  bar.fasta=extract(foo.gb, 'CDS:/gene="thrC", CDS:/gene="thrL"')</pre>	When extracting matching features, a qualifier can be optionally specified. If no qualifier is specified, the result includes all features of the given type. If a qualifier is specified, the result includes features that include a matching qualifier.

			<p><b>Note:</b> Writing many small extractions to a file format supporting features can be slow. For each extraction, all of the features are evaluated for intersection.</p>
Extract matching features using wildcards		<pre>bar.fasta=extract(foo.gb, 'CDS:/gene="thr?"')</pre>	<p>Wildcards can be used in the qualifier value. A '?' matches exactly one arbitrary character, and a '*' matches zero or more arbitrary characters. The example matches CDS features with four-character gene names beginning with "thr".</p>
Ignore source sequence qualifiers	<pre>translate(sequence_set, '/codon_start=ignore')</pre>	<pre>\$A=translate("myfile.gb", '/codon_start=ignore')</pre>	<p>Including "/codon_start=ignore" in the qualifiers causes any "/codon_start" qualifiers in the source sequence to be ignored.</p>
Translate all source sequences from DNA/RNA to protein	<pre>translate(sequence_set)</pre>	<pre>\$A=translate("myfile.fasta") \$B=translate("myfile.fasta", '/transl_table=11') Test.gb=translate("myfile.gb", '/transl_table=11') \$C=translate("myfile.fasta", '/transl_table=VERTM')</pre>	<p>If the input sequence is annotated, each CDS feature will be translated separately, and any translation table and/or codon_start annotations will be honored. If the input is unannotated, or contains no CDS features, the entire</p>

			<p>sequence will be translated. Note that translation only progresses with sequences with lengths that are multiples of three (i.e., codons); an “extra” base or two at the end will not be reflected in the output.</p> <p>The standard code is used as a default unless specified differently in the file or in the qualifier overrides. The first codon in a sequence is translated as a start codon, if recognized as such in the genetic code. Otherwise, the default translation is used.</p> <p>Translation of a DNA or RNA sequence with ambiguities might result in an amino sequence with ambiguities, or not. For example, the result of translate (“RAT”) is the ambiguity “B”, but the result of translate (“ACN”) is “T”.</p>
Override defaults or values	<pre>translate(sequence_se t, '/tag=value [,/tag=v alue...]')</pre>	<pre>\$A=translate("myfile.gb", '/transl_table=11,/codon_sta rt=ignore')</pre>	This argument is a comma-separated list of qualifiers.



specified in the file			
<p>Mark the sequences in a set as being DNA, RNA or protein.</p>	<p><code>dna(sequence-set)</code></p> <p><code>rna(sequence-set)</code></p> <p><code>protein(sequence-set)</code></p>	<pre>example.gb=protein("myfile.fasta")  bar.fasta=protein("myfile.fasta")  ("NAN",rend)</pre>	<p>This ability may be useful for sequences originating in formats where type is unspecified, such as FASTA files.</p> <p>If you use any of these functions, the information will be added to the output file, where allowed (e.g., <i>.gbk</i>). The presence of sequence type information may affect the results of searching in an endpoint expression. For example, in DNA, “N”=anything, whereas in protein, “N”=asparagine.</p>
<p>Mark the sequences in a set as being circular or linear.</p>	<p><code>circular(sequence-set)</code></p> <p><code>linear(sequence-set)</code></p>	<pre>\$A=circular(myfile.fasta)  \$B=linear(myfile.gb)  myfile.gb=circular(myfile.fasta) ("TAG",lend+4)  myfile.gb=circular("ATG"+\$B+"TAG")</pre>	<p>This ability may be useful for sequences originating in formats where type is unspecified, such as FASTA files.</p> <p>If you use either of these functions, the information will be added to the output file, where allowed (e.g., <i>.gbk</i>). The presence of sequence type information may affect the results of searching in an</p>

			<p>endpoint expression. For example, a match can cross the origin in a circular sequence, but not in a linear one.</p>
<p>Collect multiple sequence-sets into one</p>	<p><code>collect(sources)</code> (specified either as individual arguments or as single-quoted file patterns)</p>	<pre>\$a=collect(foo.fasta, bar.fasta, baz.fasta)</pre> <pre>\$a=collect("KSLQQLLLE", "ARTQTAR", "RPKPLVDP")</pre> <pre>\$a=collect('C:/MyFolder/*.fasta') &lt;br&gt;&lt;br&gt; collect('*.gb', '*.genbank')</pre>	<p>This function accepts one or more arguments, each of which may be a sequence expression or a file pattern. A file pattern is a single-quoted search string that can match zero or more file paths. Wildcards (asterisks) may be used in the filename part of this string.</p> <p>This function allows:</p> <ul style="list-style-type: none"> <li>• Multi-sequence files to be defined within a script.</li> <li>• Multiple literals or files to be used in the function argument.</li> <li>• A custom subset of the sequences in a file, by collecting individual sequences within it.</li> </ul>

			<ul style="list-style-type: none"> <li>• A custom set to be defined once in a script and then re-used later.</li> <li>• Filename patterns to be combined so as to refer to filenames with multiple extensions (see lower-most example at left).</li> </ul> <p>Note that it is possible for a sequence to appear more than once in the resulting output.</p>
Strip specified data out of sequences in a set	strip (sequence expression) (named arguments)+	<pre>strip("foo.gb")  strip("foo.gb", data='features')  strip("foo.gb", data='features,comments')  strip("foo.gb", features='CD S')  strip("foo.gb", features!='CDS')  strip("foo.gb", features='CD S, gene')</pre>	<p>If no arguments are provided, the function strips the sequences in the argument down to their name and residues. Most meta-information is stripped, including features and comments.</p> <p>Arguments:</p> <ul style="list-style-type: none"> <li>• data – Types of data to</li> </ul>

		<pre>strip("foo.gb", features='CD S:/gene=yaaA')</pre>	<p>exclude entirely.</p> <ul style="list-style-type: none"> <li><b>features</b> – Features to exclude, specified with the same syntax as the <code>feature-matching</code> argument in the <code>extract()</code> function. Additionally, the pattern is negated when the operator is <code>!=</code>. For example, the argument <code>features!='CD S,gene'</code> means to strip everything <i>except</i> for CDS and gene features.</li> </ul>
Annotate sequences in the first argument with features obtained from a <code>from*</code> argument.	<pre>annotate (sequence expression)+(named arguments)+</pre>	<pre>annotate( to.fasta, fromFile='myfile.vcf' )  annotate( to.fasta, fromFile='myfile.vcf', ids='chr1,chr2,chr3,chrX' )  annotate( to.fasta, fromSequences=from.gb )  annotate( to.gb, fromSequences=from.gb )</pre>	<p>Arguments:</p> <ul style="list-style-type: none"> <li><b>fromFile</b> – Features are obtained from a features file. Supported formats include <i>.vcf</i> (variant call format) and <i>.starff</i> (SeqNinja feature files).</li> </ul>

		<pre> annotate( to.gb, fromSequences=from.gb, features='CDS' )  annotate( to.gb, fromSequences=from.gb, features!='CDS' )  annotate( to.gb, fromSequences=from.gb, features='CDS:/gene=yaaA' ) </pre>	<p>Within the features file, the features for a particular sequence must be in a contiguous block. Blocks of features can be in any order in the file; they do not have to be in the same order as the sequences provided in the first argument. The <code>ids</code> argument (below) is necessary when the identification of a sequence differs between the sequence file and the feature file.</p> <ul style="list-style-type: none"> <li>• <code>fromSequences</code> – Features are obtained from sequences parallel and aligned to those in the first set.</li> <li>• <code>features</code> –</li> </ul>
--	--	---	--

			<p>[optional] Features to include, specified with the same syntax as the feature-matching argument in the <code>extract()</code> function. The pattern is negated when the operator is <code>!=</code>. For example, the argument <code>features!='CD S,gene'</code> means to copy everything <i>except</i> for CDS and gene features.</p> <ul style="list-style-type: none"> <li>• <code>ids</code> – [optional] Necessary with the <code>from File</code> when the identification of a sequence differs between the sequence file and the feature file. Its value is a list of the sequence IDs</li> </ul>
--	--	--	--

			in the feature file, in the order of the sequences from the first argument.
Sample the sequences in the input set	<pre>sample(sequence-set, argument=value)</pre> <p>(See description of arguments, below)</p>	<pre>sample("foo.fasta", from=10000, to=20000, by=10)</pre> <pre>sample("foo.fasta", p='0.95')</pre> <pre>sample("foo.fasta", name='GEK*')</pre>	<p>This can be useful for separating reads into different sets, or for reducing a very large number of reads to a smaller number (e.g., because of software limitations).</p> <p>Each of the arguments is optional. Any combination of arguments can be used, in any order. At most, one of each argument may be used.</p> <p>The output sequences are in the same order in which they appear in the original set.</p> <p>Specify sampling everything other than the specified value, precede the value with an exclamation mark !.</p>
	<p>Arguments for the expression 'sample':</p> <ul style="list-style-type: none"> <li>• <code>from</code> – defines the inclusive lower bound of the included sequences.</li> <li>• <code>to</code> – defines the inclusive upper bound of the included sequences.</li> </ul>		

- `by` – includes only every  $n^{\text{th}}$  element.
- `p` – is the probability that each element will be included. Calculations are made separately for each element. The elements chosen can differ between executions. The number of elements chosen is not deterministic. For large numbers of sequences, it is likely close to  $[p * \text{number of sequences}]$ .
- `name` – specifies matching sequence names, with a single-quoted string that may include an asterisk `*` as a wildcard. To exclude specific sequence names, you may use the alternate operator `!`. For example, `sample("seq.gb", name!="rat*")` includes all sequences whose name does not start with “rat”.
- `minLength` – integral expression specifying the minimum length of matching sequences, inclusive.
- `maxLength` – integral expression specifying the maximum length of matching sequences, inclusive.
- `contains` – matches any sequence on either strand that contains the single-quoted value, which may contain ambiguity codes.
- `startsWith` – sequence begins with the specified characters. If the sequence type might be DNA, a match can occur on either strand.
- `endsWith` – sequence ends with the specified characters. If the sequence type might be DNA, a match can occur on either strand.



# Actions

Miscellaneous actions that can be performed from within the SeqNinja shell include:

Objective	Expression	Examples	Comments
Separate sequences into multiple files	<code>split(sequence expression, argument=value)</code>	<pre>split(foo.fasta, to='*.fasta')</pre> <pre>split(foo.fasta, to='foo_*.fasta')</pre> <pre>split(foo.fasta, to='C:/data/foo_*.fasta')</pre> <pre>split(foo.fasta, to='*.fasta', limit=2000)</pre>	<p>This action separates each sequence into its own file. The asterisk in the filename is replaced by the sequence name.</p> <p>Allowable arguments:</p> <p><code>to</code> – (required) Specifies a filename pattern for the destination file paths. It must be a single-quoted string with an asterisk in the filename. The pattern may have a relative or absolute path. Prefixes and/or suffixes can be included in the filename.</p> <p><code>limit</code> – (optional) Limits the number of sequence files that can be generated. Otherwise, accidental specification of a file containing millions of reads could stress some operating systems. The default limit is 1000 sequence files. Note that if <code>!conversion=full</code> is specified, each sequence file may be accompanied by auxiliary files.</p>
Exit an interactive session of SeqNinja	<pre>exit</pre> <pre>quit</pre>	<pre>exit</pre> <pre>quit</pre>	

Display information in the terminal or command shell	<code>print(arguments)</code>	<code>print('Sequence=', \$myseq, 'length=', %len)</code>	<p>Accepts an arbitrary number of arguments, each of which is printed to the standard output stream. An extra space is included between the output of each argument automatically, with the exception of single-quoted string literals.</p> <p>See <a href="#">Escape Codes</a> for information on how to use character escape codes together with <code>print</code>.</p>
Add a timestamp	<code>timestamp(optional arguments)</code>	<code>timestamp('Sequence=', \$myseq, 'length=', %len)</code>  <code>timestamp</code>	<p>Same as <code>print</code>, but also adds a timestamp. Note that <code>timestamp</code> can also be invoked without any arguments, in which case it simply returns the current time.</p>
Learn how to view help	<code>help</code>	<code>help</code>	<p>Opens the file <i>README.txt</i>, which notes that SeqNinja Command-Line help can be viewed from within the Navigator and the GUI version of SeqNinja.</p>
Write data from sequences to the file specified in the to argument	<code>write(sequence expression, to=file, optional arguments)</code>	<code>write("foo.fasta", to='out.fasta')</code>  <code>write("foo.fasta", to='out.fasta', append='true')</code> <code>)</code> <code>write(\$S, to='out.vcf', template='in.vcf')</code>  <code>write("foo.gb", to='out.starff', sortFeaturesBy='type')</code>	<p>This extends the functionality provided by the file assignment syntax. It adds the ability to write to feature files and control feature order.</p> <p>Arguments:</p> <ul style="list-style-type: none"> <li><code>to</code> – [required] File into which to write the desired data. Must be a</li> </ul>

			<p>recognized file type.</p> <ul style="list-style-type: none"> <li>• <code>append</code> – True means the data should be appended to the file.</li> <li>• <code>template</code> – A file of the same type from which a template can be obtained. For VCF files, for example, this allows re-use of the file header and columns.</li> <li>• <code>sortFeaturesBy</code> – [optional] To control the order in which features are written to the destination sequence or features file. If not specified, features will be sorted into a default order. The default order sorts first by position, then by a default order of feature types. (The same default order is also affected by the file-assignment syntax.) The argument value is a comma-separated list of supported sort keys. Any of the sort keys may be preceded by a + or – to indicate ascending or</li> </ul>
--	--	--	---

			<p>descending order; the default is ascending.</p> <p>Values for <code>sortFeaturesBy</code>:</p> <ul style="list-style-type: none"> <li>◦ <code>''</code> – Two single-quote characters (denoting an empty string) are used to turn off sorting. This can increase speed when there are many features, and no final sort is required.</li> <li>◦ <code>default</code> – Explicitly specifies the default order.</li> <li>◦ <code>position</code> – Sorts by position.</li> <li>◦ <code>type</code> – Sorts alphabetically by type.</li> <li>◦ <code>/qualifier</code> – Sorts alphabetically by the values of the specified qualifier.</li> <li>◦ <code>position, type</code> – Sorts by position first, then</li> </ul>
--	--	--	---

			<p>alphabetically by feature type.</p> <ul style="list-style-type: none"><li>◦ <code>position, - type</code> – Sorts by position first, then in descending alphabetical order by feature type.</li></ul>
--	--	--	--

# Convert Between Annotated and Featureless Formats

The default `!conversion` setting of `natural` causes SeqNinja to write/read only those items that are “natural” to the receiving/source file type. Changing the setting to `full` when changing from an annotated format to a featureless format and back again creates auxiliary file(s) to preserve features and comments that would otherwise be lost.



**Note:** The conversion setting must appear in the script *before* the operation in which file output is specified.

## Convert from GenBank and Lasergene (.seq) format to FASTA format:

The setting `full` can preserve features and comments that would normally be lost when converting from annotated format to FASTA format, then back again. However, `full` is not suggested in all cases; for instance, it is not recommended when you plan to make sequence modifications (other than substitutions) while the sequence is in FASTA format. When `full` is selected, SeqNinja outputs three files during conversion from an annotated format to FASTA format:

- *.fasta* – This is a standard FASTA file containing sequence reads. When writing FASTA files from a source that contains comments, the FASTA header line includes the first line of comments from the source. For example:  
  

```
> NC_010473 LOCUS NC_010473 4686137 bp DNA circular BCT 04-OCT-2012
```
- *.starff* – This file contains tab-delimited feature information from the original annotated file and can be opened and modified using a spreadsheet utility such as Microsoft Excel®. The modified file can be read by SeqNinja, with the following restrictions: 1) Sequences must appear in exactly the same order as in the corresponding FASTA file. 2) The first six columns must remain present in the original order. Subsequent columns contain qualifier values named in the column header, and can be removed or amended, as desired. 3) The modified file must be saved as tab-delimited text, not as an .xls or .xlsx file, and its name must remain unchanged.
- *.fasta.comments* – This file contains one comment pertaining to each sequence in the FASTA file, and in the same order. Comments are maintained during file conversions (e.g., `a.seq = b.gbk`). However, comments are discarded during sequence transformations (e.g., `a.gbk = b.gbk + c.gbk`).

## Convert the saved FASTA file back to GenBank or Lasergene format:

If you used `full` during the conversion to FASTA, you can convert back to GenBank or Lasergene, using `full` again, without losing the original features and comments. As long as you keep the auxiliary files in the same folder as the FASTA data file, SeqNinja will automatically place the features and comments back in the GenBank or Lasergene file.

# Use Setting Substitution

SeqNinja allows you to substitute the values of settings into single- and double quoted strings. You can substitute the value of any setting, including built-in settings, custom settings, and command-line arguments. When you run the script, SeqNinja will replace the name of the setting with its value. For example, a string (e.g., a file path) or a numeric setting (e.g., an integer) will be used in place of the setting name. This functionality allows the same script to be reused with different data and allows the script to be moved along with its data.

Substitution involves several steps, each of which is discussed below.

## Step 1 – Enabling substitution:

Before substitution is used in a script, you must first enable “substitution” in that script. This is done by adding `!setting:substitution = 'true'` anywhere in the script above the substitution.

## Step 2 – Defining a setting:

Settings can be defined...

- in the script.
- as an argument (e.g., `!{1}`).
- implicitly (see the “Tip” below for an example).

The syntax for a user setting is `!user:<optional text> = <value> ...` where:


<code>!user:</code>	Recommended at the beginning of each custom setting, to avoid name clash with current and future built-in settings.
<code>optional text</code>	Used to differentiate settings or to remind you of the purpose of a setting. For example, <code>user:tags</code> , <code>user:fasta</code> , <code>user:genbank</code> .
<code>value</code>	The information you want to substitute for the setting when the script is run (e.g., a file path, integer, etc.). Substitution can occur in numeric expressions or in strings. Strings are sequences of characters (e.g., a file path) between pairs of single or double quotes.

## Step 3 – Specifying substitution:

Curly braces are used to substitute the value of a setting in a string or numeric expression.



Arguments can be referenced in the script with `!{1}`, `!{2}`, etc. The path to the script file itself is `!{0}`. To refer to a '!' in the script, use `!!`.

 **Tip:** To refer to the directory containing the script (if any), use `!{script:dir}`. An earlier version of this command, `!{program:dir}` is also supported. This allows scripts to refer to data in the same directory, even if the directory is moved or if SeqNinja is launched from a different location.

#### Examples using custom settings:

**Example 1:** Write and save a script called *convert.star*. Then issue commands like the ones below:

```
C:\> seqninja.exe convert.star mg1655.gb mg1655.fasta
C:\> seqninja.exe convert.star DH10B.fasta DH10B.gb
C:\> seqninja.exe convert.star adh1_human.pro adh1_human.fap
```

**Example 2:** Write and save a script called *extractCDS.star*. Then issue commands like the ones below:

```
C:\> seqninja.exe extractCDS.star ref_chr2L
C:\> seqninja.exe extractCDS.star ref_chr3L
C:\> seqninja.exe extractCDS.star ref_chr4L
```

#### **Context-specific examples:**

Context	Example	Comment
<b>Paths</b>	"out.fap" = "!{program:dir}/adh4_mouse.pro"	
<b>Sequence literals</b>	"out.fasta" = "!{1}" + "in.fasta"	In this example, the user passes a sequence of residues as the first command-line argument. For literals defined in the script, it's easier just to use a sequence variable:  \$INSERTION = "NNNNNNNN" "out.fasta" = \$INSERTION + "in.fasta"
<b>Patterns</b>	"out.fasta" = collect( '!{program:dir}/genomes/ Vibrio*.gbk' )	
<b>Sequence index</b>	"out.fasta" = "454.fasta#"!{1}"	
<b>Settings</b>	!mydir="C:\Users\norville"	

	!mydata="!{mydir}/data" !myresults="!{mydir}/results"	
<b>Integers</b>	out.fasta = in.gb ( %!{1} > %!{2} )	<p>This allows using integers passed on the command line in numeric expressions. For values defined in the script, it's easier just to use variable integers:</p> <pre>%A = 42 %B = 111 out.fasta = in.gb( %A &gt; %B )</pre>
<b>Single-quoted strings</b>	print( 'File: !{1}' )	

# Edit in the SeqNinja Shell

---

When using the SeqNinja shell in Macintosh, editing shortcuts such as **Cmd+A** (select all) and **Cmd+V** (paste) work as expected. When using the SeqNinja shell in Windows, however, these shortcuts are disabled. To access editing commands from the shell in Windows, right-click on the header bar at the top of the command-line window and choose **Edit > (command-name)**.

- **Edit > Select All** – Selects all text in the current command-line session.
- **Edit > Paste** – Pastes the contents of the clipboard in the current line.
- **Edit > Find** – Launches the Find dialog.

# Perform Multiple Calculations

---

When performing multiple calculations, the steps must be set up in a particular way. For example, if the input file for one step is the output file from an earlier step, you may obtain unexpected results. Specifically, the final output may have additional sequences and/or features.

Script A is an example of a workflow that can lead to such unexpected results:

## SCRIPT A

- Step 1: Extract features from TEST.GBK and save them as EXTRACTED.GBK.
- Step 2: Translate EXTRACTED.GBK and save the results as TRANSLATED.GBK.
- Result: The file TRANSLATED.GBK may contain unexpected data (e.g., an extra translated feature).

This situation arises when there are overlapping CDS features. In such cases, a piece of one CDS will end up being annotated in the interval it shares with the second CDS. Then, when the EXTRACTED.GBK file is translated, that particular sequence will result in two protein sequences: the desired full length CDS, and the fragment of the overlapping CDS. Note that this is not an issue when the intermediate file is in FASTA format since there is no carryover of information about the overlapping CDS.

If you need the extracted CDS sequences to be in GenBank format instead of FASTA format, you should instead use the procedure described in Script B:

## SCRIPT B

- Step 1: In the SeqNinja shell, enter the following:
 

```
:: !conversion=full
:: extracted_CDS.fas = extract(genome.gbk, 'CDS')
```
- Step 2: Leaving the SeqNinja shell open, open the resulting *.starff* file using any text editor and delete extraneous annotation lines. Save the updates.
- Step 3: In the SeqNinja shell, enter the following:
 

```
:: extracted_CDS.gbk = extracted_CDS.fas
:: translated_CDS.gbk = translate(extracted_CDS.gbk)
```
- Result: Overlapping fragments of CDS features are absent from the Genbank-formatted output file, which is the desired outcome.

# Example Scripts

---

The following scripts are intended only as a starting point, and must be adapted, as necessary, for your particular data and goals.

If you are using the [graphic user interface version of SeqNinja](#), you can copy the text in the **Script** row of each example paste into the [Text view](#). When you switch to the [Visual view](#), it will be populated with the example, helping guide you on which templates and options to choose for your own setup.

- [Add segments of literal sequence to every sequences in a set](#)
- [Collect sequences with certain attributes into a single file](#)
- [Concatenate and de-concatenate contigs for gene finding](#)
- [Convert one version of a sequence to another](#)
- [Divide a genome into pieces of a specified length & with a specified overlap](#)
- [Extract annotated features from a genome](#)
- [Extract segments from a sequence](#)
- [Extract a specific set of genes or proteins from multiple genomes](#)
- [Insert a missing base into a sequence](#)
- [Reverse complement a sequence](#)
- [Split a multi-sequence file into individual files](#)

# Add segments of literal sequence to every sequence in a set

---

<b>Goal</b>	To add "ATG" at the beginning and "TAG" at the end of every sequence in a multi-sequence file named <i>two.fasta</i> . The file <i>two.fasta</i> contains two sequences:  aaact aaactct
<b>Script</b>	<pre>print("ATG" + two.fasta + "TAG")</pre>
<b>Output</b>	ATGaaactTAG ATGaaactctTAG

# Collect sequences with certain attributes into a single file

---

<b>Goal</b>	The input directory <i>C:\data</i> contains the files <i>alpha.fas</i> , <i>alphabet.fasta</i> , <i>alpha.fap</i> , and <i>beta.fas</i> . The goal is to collect all the sequences in the directory whose sequence sets begin with “alpha,” and whose extensions begin with “.fas,” and write them to the file <i>results.fasta</i> .
<b>Script</b>	<pre>!datapath = "C:\data" "results.fasta" = collect('alpha*.fas*')</pre>
<b>Output</b>	Collects only the sequences from sequence sets <i>alpha.fas</i> and <i>alphabet.fasta</i> and saves them as a multi-sequence file named <i>results.fasta</i> .

The output can be split back into multiple files using the procedure in [Split a multi-sequence file into individual files](#).

# Concatenate and de-concatenate contigs for gene finding

---

<b>Goal 1</b>	To merge contigs into a pseudo-molecule with a 36 bp linker that introduces a stop codon (linker: NNNNNCACACACTTAATTAATTAAGTGTGTGNNNNN) in all six frames.
<b>Script 1</b>	<pre>\$linker="NNNNNCACACACTTAATTAATTAAGTGTGTGNNNNN"  pseudo.fas=contig00001.fas+\$linker+contig00002.fas</pre>
<b>Goal 2</b>	<p>After annotation or gene-finding via <i>pseudo.gbk</i>, break the pseudo-molecule back into contigs:</p> <p>contig00001 length=116581</p> <p>contig00002 length=97153</p> <p>You will need to determine the correct endpoints for the de-concatenation step, as SeqNinja cannot do this automatically.</p>
<b>Script 2</b>	<pre>contig00001.gbk=pseudo.gbk(1,116581)  contig00002.gbk=pseudo.gbk(116618,213770)</pre>



# Convert one version of a sequence to another

The following scripts can be used to convert genome coordinates between assemblies or to migrate annotations between different versions of a genome while preserving annotations.

In this example, researchers identified seven mutations in *E. coli* strains that did not appear in the reference sequence. They wanted to determine the impact of those changes on the features annotated in the genome. Therefore, they needed to migrate annotations between different versions of a genome while simultaneously converting genome coordinates between assemblies.

<b>Goal 1</b>	<p>To convert the reference sequence into the mutant sequences with full annotation. The following changes are required:</p> <ul style="list-style-type: none"> <li>• 1 bp SNP “A” to “G”</li> <li>• insert “G” (duplicate preceding “G”)</li> <li>• 1 bp SNP “C” to “T”</li> </ul>
<b>Script 1</b>	<pre>Mutant_A.gbk=Reference_Seq.gbk(1,547693)+"G"+ Reference_Seq.gbk(547695,547832)+Reference_Seq.gbk (547832,3957956)+"T"+Reference_seq.gbk(3957958,rend)</pre>
<b>Goal 2</b>	<p>To convert the annotated reference sequence to the mutant-B sequence, making the following changes:</p> <ul style="list-style-type: none"> <li>• &lt; IS1 &lt; insertion + 8 bp target duplication</li> <li>• 1 bp SNP “A” to “G”</li> <li>• insert “G” (duplicate preceding “G”)</li> <li>• &gt; IS5 &gt; insertion + 4 bp target duplication</li> <li>• insert “CC”</li> <li>• 1 bp deletion/frameshift (-G)</li> <li>• 1 bp SNP “C” to “T”</li> </ul>
<b>Script 2</b>	<pre>Mutant_B.gbk=Reference_seq.gbk(1,257907)+complement(IS1.fas) +Reference_seq.gbk(257900,547693)+"G"+Reference_seq.gbk(547695,547832)</pre>

```
+Reference_seq.gbk(547832,1298721)+IS5.fas+Reference_seq.gbk(1298718,2171386)
+"CC"+Reference_seq.gbk(2171387,3558477)+Reference_seq.gbk(3558479,3957956)
+"T"+Reference_seq.gbk(3957958,rend)
```

# Divide a genome into pieces of a specified length & with a specified overlap

---

<b>Goal</b>	To break up a genome into pieces of length $N$ with overlap $m$ (e.g., for simulations).
<b>Script</b>	<code>pieces.fas=cut(U00096.2.gbk,240,60)</code>
<b>Output</b>	25775 sequences of 240 bp, overlapping by 60 bp, plus the final part of the sequence.

# Extract annotated features from a genome

This example can be adapted to extract relevant annotated features (e.g., specific feature types or all features with specified annotations) for uses such as building BLAST databases or consensus matrices or performing alignments. Script A generates an overlapping CDS (specifically, the yeaC fragment). Using Scripts B or C solves this problem. Scripts B and C produce identical results, but only B also outputs the nucleotide sequence file.

<b>Goal</b>	To extract a set of annotated CDS features from a genome as protein sequences
<b>Script A</b>	<pre>m54sCDS.gbk=extract(m54s.gbk, 'CDS')  m54s_proteins1.fas=translate("m54sCDS.gbk")</pre>
<b>Output A</b>	<p>LOCUS U00096:yeaA 414 bp DNA 13-JAN-2012</p> <p>FEATURES Location/Qualifiers</p> <p>Source 1..414</p> <p>Source complement (1..414)</p> <p>/seqninja_feature_id="0000000001//Users/guy/Desktop/seqninja_testing/m54s.gbk:1"</p> <p>/note="***Needs review***Cut segment head by 1860039 and tail by 2778768 units."</p> <p>/organism="Escherichia coli"</p> <p>CDS 411..414</p> <p>/gene="yeaC"</p> <p>seqninja_feature_id="0000000012//Users/guy/Desktop/seqninja_testing/m54s.gbk:1"</p> <p>/note="***Needs review***Cut segment tail by 314 units."</p> <p>CDS 1..414</p> <p>/gene="yeaA"</p> <p>/seqninja_feature_id="0000000013//Users/guy/Desktop/seqninja_testing/m54s.gbk:1"</p>

	<p>ORIGIN</p> <p>1 atggctaata aaccttcggc agaagaactg aaaaaaaatt tgtccgagat gcagttttac  61 gtgacgcaga atcatgggac agaaccgcca ttacgggtc gttactgca taacaagcgt  121 gacggcgtat atcactgttt gatctgcgat gccccgctgt ttcattccca aaccaagtat  181 gattccggct gtggctggcc cagtttctac gaaccggtaa gtgaagaatc cattcgttat  241 atcaaagact tgcacatgg aatgcagcgc atagaaattc gttgcggtaa ctgtgatgcc...</p>
<b>Script B</b>	<code>m54sCDS.fas=extract(m54s.gbk, 'CDS')m54s_proteins2.fas=translate("m54sCDS.fas", '/transl_table=11')</code>
<b>Script C</b>	<code>m54s_proteins3.fas=translate("m54s.gbk")</code>

# Extract segments from a sequence

Goal	To extract the C-regions (exons), J-segments and V-segments from this T-receptor locus sequence (AF159056)
Script	<code>4Matt.fas=extract (AF159056.gbk, 'V_segment,J_segment,C_region')</code>
Output ( <i>Matt.fas</i> )	<pre>&gt;Matt:1 atgcggtgggcccctagcgggtgcttctagctttcctgtctcctggtgagtgcgctgcctacagagaggatcacggggtttgtttgtttgtattttcttctttgcaagg &gt;Matt:2 atgcagtgggcccctagcgggtgcttctagctttcctgtctcctggtgagtgcgctgcctacagagaggatcacggggtttgtttgtttgtattttcttctttgcaaggagtacc &gt;Matt:3-25(similar; omitted here to conserve space) ... &gt;Matt:26 atacactactgctgcagctcacaacacctctgcatattacatgtacctcctcctgctcctcaagagtgtgggtctatttgccatcatcacctgctgtctgcttggag</pre>

# Extract a specific set of genes or proteins from multiple genomes

---

<b>Goal</b>	First, to generate a multiple-genome file called “all.gbk” from all the .gbk files in the directory “genomes”. Second, to extract a specific set of genes or proteins from “all.gbk.”
<b>Script</b>	<pre>!datapath:in = "/Users/[UserName]/Desktop/SN_test/genomes"!datapath:out = "/Users/[UserName]/Desktop/SN_test""all.gbk" = collect( '*.gbk' ) "MLST.fasta" s" = extract( "/Users/[UserName]/Desktop/SN_test/all.gbk", 'CDS:/gene="adk",CDS:/gene="fumC",CDS:/gene="gyrB",CDS:/gene="icd*",CDS:/gene="mdh",CDS:/gene="purA",CDS:/gene="recA"' )</pre>

It is necessary to either have a different datapath:in and datapath:out at the point of the collect step, or to use the full path to all.gbk. Otherwise, the result will be an endless loop.

# Insert a missing base into a sequence

---

<b>Goal</b>	To insert a “G” between 39352 & 39353 in an existing sequence.
<b>Script</b>	<code>newseq.gbk=oldseq.gbk(lend, 39352)+"G"+oldseq.gbk(39353, rend)</code>



# Reverse complement a sequence

---

<b>Goal</b>	To reverse complement (flip) 4750 reads in the file MID3_13.fas
<b>Script</b>	<code>"rcMID3_13.fas"=~ ("MID3_13.fas")</code>

# Split a multi-sequence file into individual files

<b>Goal</b>	To split a multi-sequence file into individual files according to specified criteria.
<b>Script</b>	<code>split("results.fasta", to='alpha_*.fas')</code>
<b>Output</b>	The two files ( <i>alpha.fas</i> and <i>alphabet.fasta</i> ) comprising the multi-sequence <i>results.fasta</i> file are saved as separate sequence files using the names in their definition lines, prepended by “alpha_”.

# Appendix

---

The following topics are applicable to both the GUI and command-line versions of SeqNinja.

- [Supported file types](#)
- IUPAC codes for [amino acids](#) / [nucleotides](#)
- [Create a script to use as MegAlign Pro input](#)
- [Research references](#)

# Supported file types

SeqNinja can both read and write to any of the following file formats:

File type	Extensions recognized	Comments
FASTA single and multi- sequence files	<i>.fasta</i> , <i>.fas</i> , <i>.fna</i> , <i>.fap</i>	For more information, see NCBI's Accepted Input Formats page.
GenBank single and multi- sequence files.	<i>.genbank</i> , <i>.gb</i> , <i>.gbk</i> , <i>.genpept</i> , <i>.gp</i>	In a GenBank multi-sequence file, each sequence is terminated by a double slash sign (//). For more information about GenBank flat file format, see NCBI's GenBank Flat File page.
SeqNinja set files	<i>.star</i>	SeqNinja can read, write, and execute set files, which are themselves editable. Sequences can be removed or re-ordered, though re-ordering sequences in large datasets can significantly slow processing time. A union of sets can be created by concatenating two set files.
DNASTAR and Lasergene sequence files	<i>.seq</i> , <i>.pro</i> , <i>.mseq</i>	Lasergene DNA and protein files have the extensions <i>.seq</i> and <i>.pro</i> , respectively. DNASTAR multiple-sequence DNA files have the extension <i>.mseq</i> , and can be created by SeqMan Ultra.

For a list of file formats supported by this DNASTAR product, please see the [Supported File Types](#) page of our website.

Note that many Lasergene applications will import sequences in multiple file formats. [Click here](#) for a list of import types by application. If a file type is not listed in your application of choice, you should open it in SeqBuilder Pro and either save it in the Lasergene sequence file format (*.seq*) or select **File > Send Sequence To** to open it in the application of your choice.

# Change the default application for opening files

---

## On Windows 10:

1. In the search bar on the bottom left of the screen, type in *Default apps* and press the **Default apps** icon.
2. Use any of the links below to set the default app for opening a particular file type.

[Choose default apps by file type](#)

[Choose default apps by protocol](#)

[Set defaults by app](#)

[Recommended browser settings](#)

## On Windows 7 and Windows 8:

1. Open the Control Panel.
2. Go to **Programs > Default Programs > Associate a file type or protocol with a specific program**.
3. Select extension you wish to change in the **Extensions** list and then click the **Change program** button.
4. Click the **Browse** button in the Open With dialog box.
5. Navigate to and select the application you want the files to open in. (Lasergene applications are located by default in *C:\Program Files\DNASTAR\Lasergene 'x'* or *C:\Program Files (x86)\DNASTAR\Lasergene 'x'*, where 'x' represents the version number).
6. Click **Open**.
7. Click **OK** to close the Open With dialog box, and then click **Close**.

## On Macintosh:

1. Select a file of the type you wish to change in Finder.

2. Go to **File > Get Info**.
3. Click the dropdown arrow in the **Open With** field, and navigate to the application you want the files to open in. (Lasergene applications are located by default in *Applications:DNASTAR:Lasergene 'x'*, where 'x' represents the version number).
4. Click the **Change All** button to make the application you selected the default for all files of this type.

# IUPAC codes for amino acids

---

1-Letter Code	3-Letter Code	Amino Acid
<b>A</b>	Ala	Alanine
<b>B</b>	Asx	Aspartic Acid or Asparagine
<b>C</b>	Cys	Cysteine
<b>D</b>	Asp	Aspartic Acid
<b>E</b>	Glu	Glutamic Acid
<b>F</b>	Phe	Phenylalanine
<b>G</b>	Gly	Glycine
<b>H</b>	His	Histidine
<b>I</b>	Ile	Isoleucine
<b>J</b>	Xle	Leucine or Isoleucine
<b>K</b>	Lys	Lysine
<b>L</b>	Leu	Leucine
<b>M</b>	Met	Methionine
<b>N</b>	Asn	Asparagine
<b>O</b>	Pyl	Pyrrolysine
<b>P</b>	Pro	Proline
<b>Q</b>	Gln	Glutamine
<b>R</b>	Arg	Arginine
<b>S</b>	Ser	Serine
<b>T</b>	Thr	Threonine
<b>U</b>	Sec	Selenocysteine
<b>V</b>	Val	Valine
<b>W</b>	Trp	Tryptophan
<b>X</b>	Xaa	Unspecified or unknown
<b>Y</b>	Tyr	Tyrosine
<b>Z</b>	Glx	Glutamic Acid or Glutamine

# IUPAC codes for nucleotides

Symbol	Description	Bases Represented			
<b>A</b>	adenosine/adenine	A			
<b>C</b>	cytidine/cytosine		C		
<b>G</b>	guanosine/guanine			G	
<b>T</b>	thymidine/thymine				T
<b>U</b>	uridine/uracil				U
<b>W</b>	weak	A			T
<b>S</b>	strong		C	G	
<b>M</b>	amino	A	C		
<b>K</b>	keto			G	T
<b>R</b>	purine	A		G	
<b>Y</b>	pyrimidine		C		T
<b>B</b>	not A		C	G	T
<b>D</b>	not C	A		G	T
<b>H</b>	not G	A	C		T
<b>V</b>	not T	A	C	G	
<b>N or –</b>	any base (not a gap)	A	C	G	T

✿ **Note:** The non-standard nucleotide codes “I” (Inosine) or “X” (xanthine) are not supported.



# Create a script to use as MegAlign Pro input

[MegAlign Pro](#) is Lasergene's application for performing multiple alignments of large DNA, RNA, or protein sequences using one of three alignment methods: MUSCLE, Mauve, or Clustal Omega. MegAlign Pro also offers several post-alignment analysis tools, including phylogenetic tree generation and access to a distance matrix, showing distances between each pair of sequences.

When you use MegAlign Pro, you need to enter the sequences you wish to align. It is possible to add a SeqNinja .star project file in lieu of the usual sequence files. The advantage of adding a .star project to MegAlign Pro is that it will always call up the most up-to-date version of the input data for use in making its output. So, if you change, add, or delete the input data that the .star file points to, the .star file will update this information in MegAlign Pro.

When you use a SeqNinja Project (.star) file as input for MegAlign Pro, note that the project is read as a "set of sequences," and is not actually executed, meaning no sequence files are written. This type of project has limitation that any input files referenced in the script must already exist. In other words, the script should not reference an input file that doesn't exist until it is created earlier in the same script. Instead:

- Replace all the references to the input file with a variable (e.g., \$a), or...
- Create a SeqNinja script to create the necessary files and run the script. Then, in the SeqNinja script you wish to add to MegAlign Pro, reference those already-created files.

When creating a SeqNinja script for use as MegAlign Pro input, you may break statements into multiple lines before or after the following operators:

,	comma
+	plus sign
=	equals sign
+=	plus equals sign

For examples, any of the following SeqNinja steps (as they appear in the [Text view](#)) would be read as a set of sequences in MegAlign Pro:

## Example 1:

```
$a = $b
+ $c
```

## Example 2:

```
$a = $b +
```

\$c

**Example 3:**

```
$a = sample("myfile.fasta",  
from=100,  
to=200)
```

# Research references

---

Schroeder JL and Blattner FR (1982). "Formal description of a DNA oriented computer language." Nucleic Acids Res 10 (1):69-84 [PMID: 7063408; PMCID: [PMC326115](#)].