MacVector 26.0

for Mac OS X

Introduction to Golden Gate Cloning

MacVector, Inc.
Software for Scientists

Copyright statement

This version of the Simple Golden Gate Cloning tutorial was published in July 2026.

Contents

Introduction

Golden Gate Cloning is a molecular cloning technique that enables the seamless, ordered assembly of multiple DNA fragments into a single construct in one reaction. Its defining feature is the use of Type IIS restriction enzymes, which recognize specific asymmetric sequences but cleave the DNA at a defined distance outside of that recognition site rather than within it. Because the cut site is separated from the recognition sequence, the actual overhangs generated are not fixed by the enzyme but can be freely designed by the researcher. By engineering complementary single-stranded overhangs (typically 4 nucleotides long) onto the ends of each fragment, you can dictate exactly which pieces ligate together and in what order. This allows digestion and ligation to occur simultaneously in a single "one-pot" reaction—an efficient, scarless, and highly modular approach that underpins standardized assembly frameworks like MoClo and the broader synthetic biology toolkit.

The most commonly used Type IIS enzymes for Golden Gate are *Bsa* I (recognition sequence GGTCTC), *Bsm* BI/*Esp* 3I (CGTCTC), and *Bbs* I/*Bpi* I (GAAGAC), each cleaving a short, fixed distance away from its recognition site to leave a 4-nucleotide overhang. The key to designing primers is to position the recognition site so that it sits on the outside of the intended 4nt overhang—that is, the enzyme cuts inward toward your fragment of interest, leaving the custom overhang on the insert while the recognition sequence remains on the flanking DNA that gets discarded upon cleavage. This orientation is what makes the reaction so efficient: once two fragments ligate through their matching overhangs, the recognition sites have been cut away and are no longer present at the new junction, so the enzyme cannot re-cut the assembled product. Correctly ligated constructs are therefore removed from the digestion-ligation equilibrium and accumulate, while any incorrect or unligated products are repeatedly re-cut and given another chance to assemble—driving the reaction strongly toward the desired final construct.

MacVector lets you create a Golden Gate Cloning “Cloning Project” to which you can add cloning vectors with appropriate Type IIS sites, fragments of DNA that you might want to clone into the vector, or even a collection of pre-determined circular constructs that could be cut with a specific Type IIS enzyme an assemble into a single construct in a one-pot reaction. Where appropriate, MacVector will generate primers to amplify fragments to generate compatible overhangs to drive the cloning. At any time, you can customize the primer to add additional residues to ensure coding sequences will be fused in the correct frame or change the chosen asymmetric overhang to one of the common recommended sequences.

Sample Files

From MacVector 26.0 onwards, you can find most of the sample files for this tutorial in the directories below;

```
/Applications/MacVector/Sample Files/
```

```
/Applications/MacVector/Tutorial Files/Golden Gate/
```

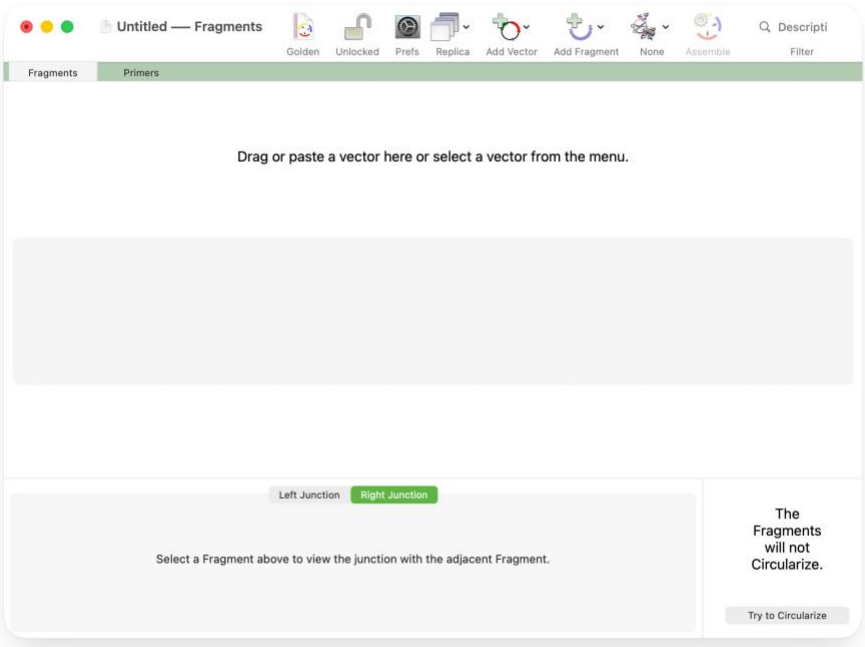
Tutorial

Overview

This simple tutorial will demonstrate the very basic capabilities of the MacVector Golden Gate Cloning interface. We will create a new project, select an enzyme, add a suitable vector and then a couple of insert fragments and then create a final construct and view the required primers. A separate Advanced Tutorial explores the more detailed and nuanced features of the Golden Gate Cloning interface.

Creating a Project

Choose **File | New | Golden Gate Cloning**. A new window will open that is set to the "**Golden Gate**" mode:



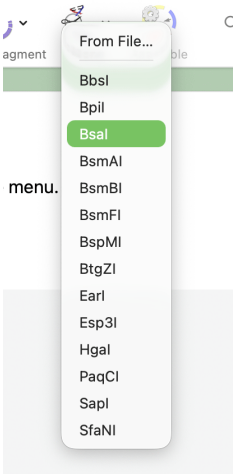
Previous versions of MacVector let you toggle between Gibson/LIC/Manual modes. But Golden Gate Cloning is sufficiently different in principle than the others (enzyme generated short overhang based, rather than long homology overhangs) that it dramatically simplifies the interface to select the technology at the initiation of the project.

Selecting an Enzyme

By default, MacVector uses a custom `Golden Gate.renz` file that contains all the Type IIS enzymes suitable for use in Golden Gate cloning. That file is in your `~/Library/Application Support/MacVector/Restriction Enzymes/` directory. You can edit this file to add (or delete) Type IIS enzymes relevant to Golden Gate cloning.

Any enzymes present in the file are displayed in the popup menu that appears when you click on the **Enzyme** toolbar button (labelled **None** in the above image, taken before any enzyme was selected).

Click on the **Enzyme** button and select *Bsa* I from the menu.



Adding a Vector

There are many ways to add a vector to a project. The interface assumes that the first DNA fragment in the list is the primary vector, but this is not absolutely required. You can add a vector by;

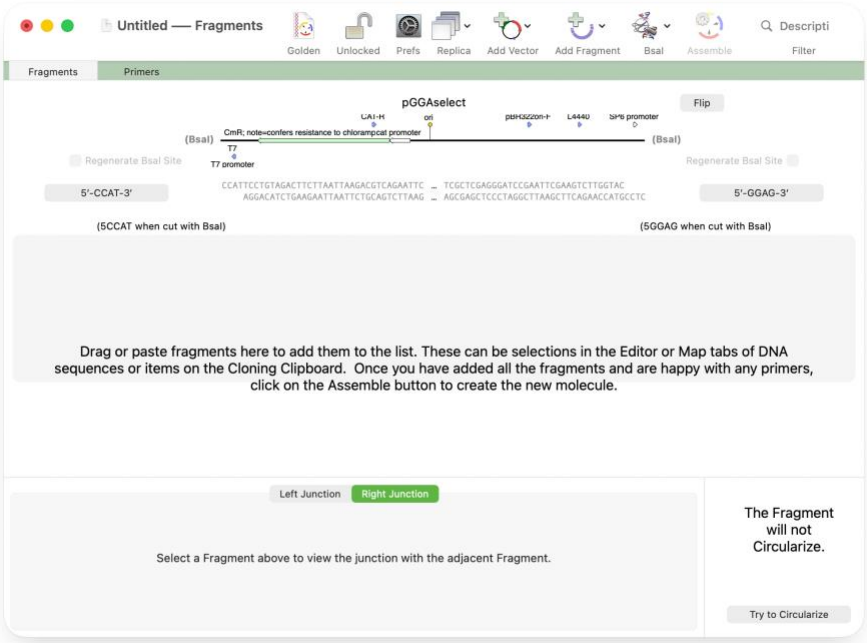
(a) Selecting a vector from the **Add Vector** button. This shows a popup menu that displays a hierarchical view of the MacVector `.nucl` files present in your `~/Library/Application Support/MacVector/Common Vectors/Golden Gate Vectors/` folder. You can add your own files/folders to this location if you wish.

(b) **Add Vector | From File**. You can select any MacVector readable file from any location on your file system and it will get added to the project.

(c) Drag and drop. Typically, you would graphically select and hold a restriction enzyme in the **Map** tab of a vector and drag that onto the project window.

In all these cases, MacVector will scan the vector for the currently selected Type IIS enzyme. If it finds two *facing* cut sites, it will virtually "cut" the vector and display the linearized vector in the first panel and display the overhangs at each end. If two appropriate sites are NOT found, it will assume you want to break the vector at either the cut site (for an restriction enzyme site drag and drop) or the circular origin (for a circular vector) and will generate primers to create suitable Golden Gate overhangs at the ends of the linearized molecule.

Click and hold on the **Add Vector** toolbar button item. In the popup menu that appears, select the **NEB | pGGASelect** menu item. The project will update to display the vector cut with *Bsa* I;



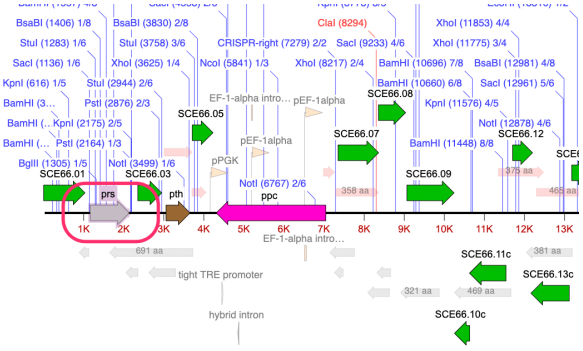
Note that MacVector has identified the *Bsa* I restriction enzyme cut sites, realized that they "face" each other and displayed the linearized fragment in the first pane.

Adding Fragments

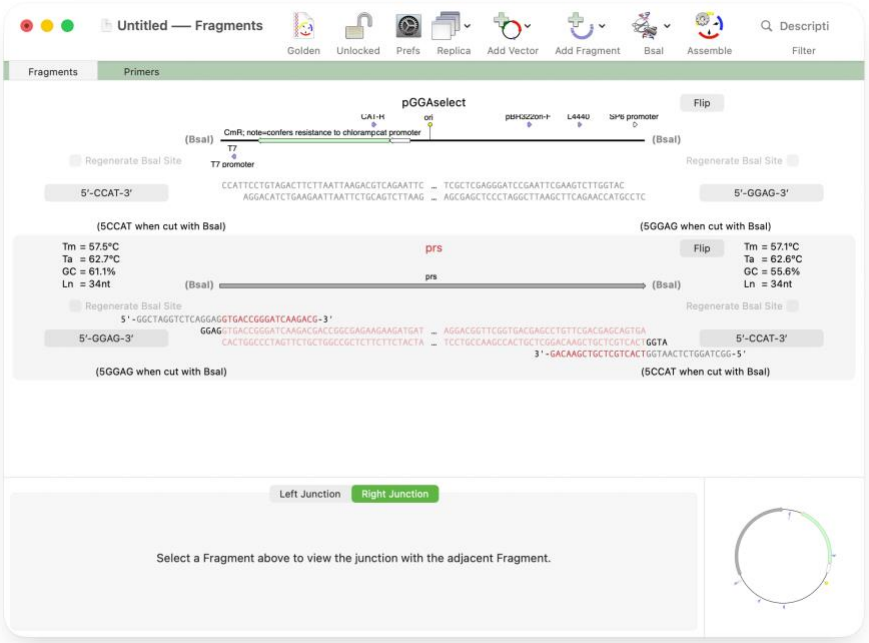
Adding fragments to the cloning project is not that much different from adding a vector. Typically, you would add them using drag and drop or copy/paste, but you can also click on the **Add Fragment | From File** toolbar button and add DNA sequences in any MacVector supported formats.

Open the `Gal Cosmid.nucl` file from `/Applications/MacVector/Sample Files/` and switch to the **Map** tab.

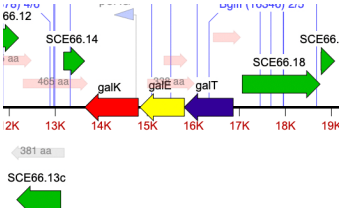
Select the grey *prs* gene and drag it to the project window.



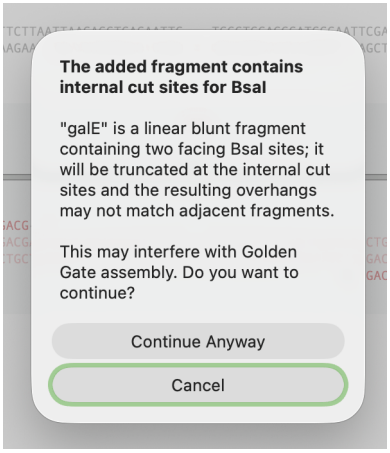
It gets added and the view updates to show the fragment and any primers required to create *Bsa* I sites with appropriate matching 4 nt overhangs via PCR..



Now drag the bright yellow *galE* gene into the window.

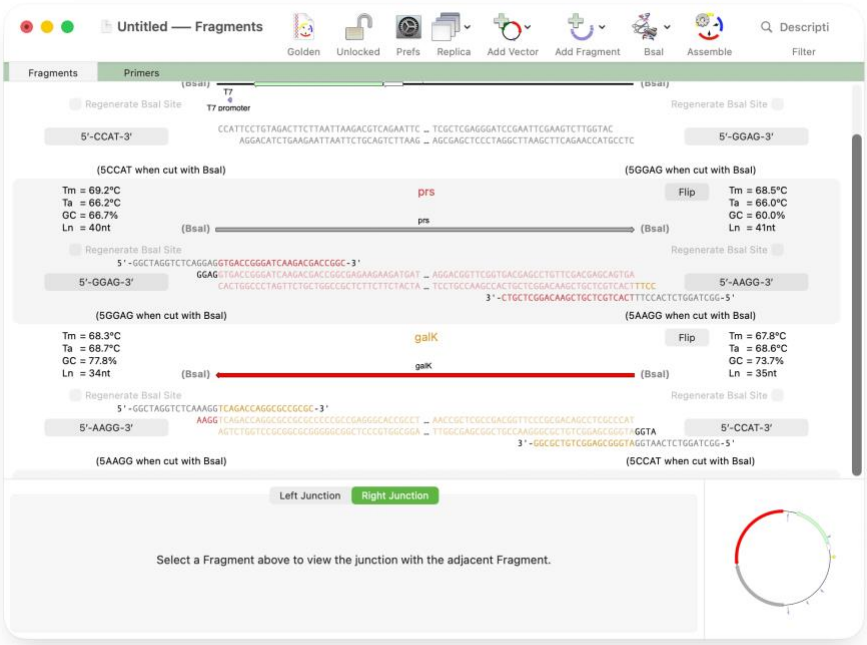


This has internal *Bsa* I sites so a warning dialog appears. Using this fragment in a *Bsa* I driven Golden Gate cloning would prevent the assembly from working as expected. You could try using a different Type IIS enzyme, or you could use site-specific mutagenesis to remove the sites, but that requires significant additional effort.

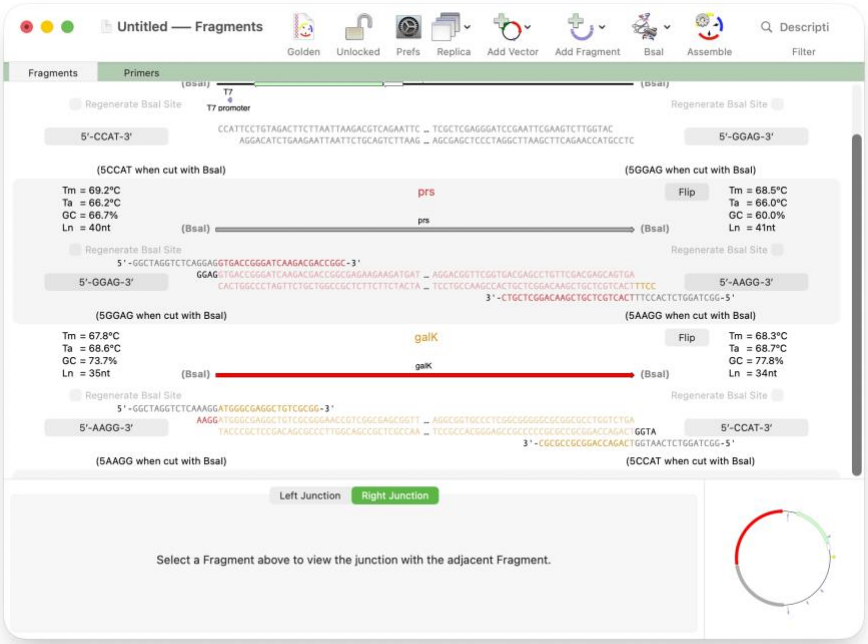


Click on **Cancel** to avoid adding the fragment.

Now select the red *galK* gene and drag it into the project window. That gets added and the ends get updated with new primers.

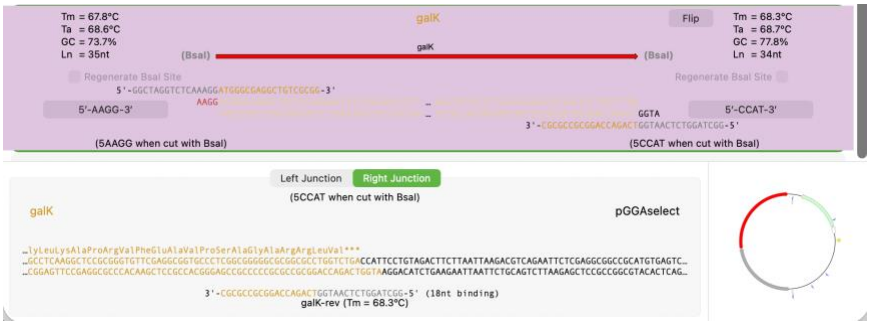


But let's add that in the other orientation. Click on the **Flip** button at the right end of the *galK* pane. The fragment reverses orientation. If you look carefully, you can just make out the miniature arrowheads at the end of the red graphic indicating the direction of translation.



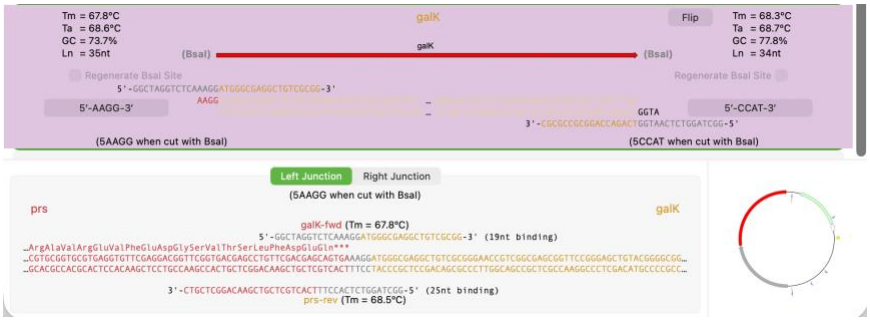
Viewing Junctions

Click on the *galK* fragment pane. The lower panel updates to show the detail of the selected end – in this case the **Right Junction**.



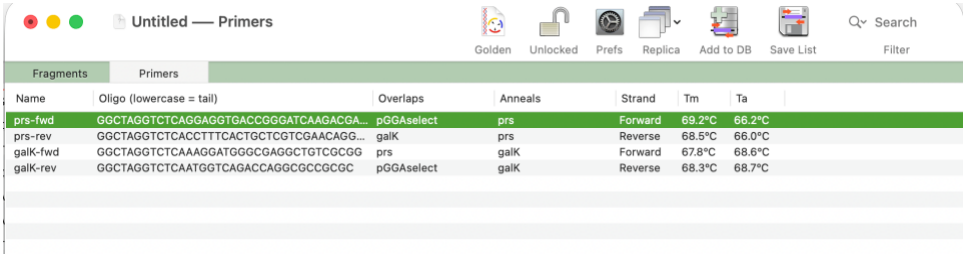
You can see the junction between the fragments where the unique *Bsa* I generated overhang joins the two fragments, along with the primer for the right end of the *galK* gene (`pGGASelect` does not require a primer) along with the translated amino acids at the end of the *galK* gene.

Similarly, clicking on the **Left Junction** tab shows the junction between the *prs* and *galK* fragments.



Viewing Primers

Click on the **Primers** tab. This displays a spreadsheet view of the primers required to generate this construct.



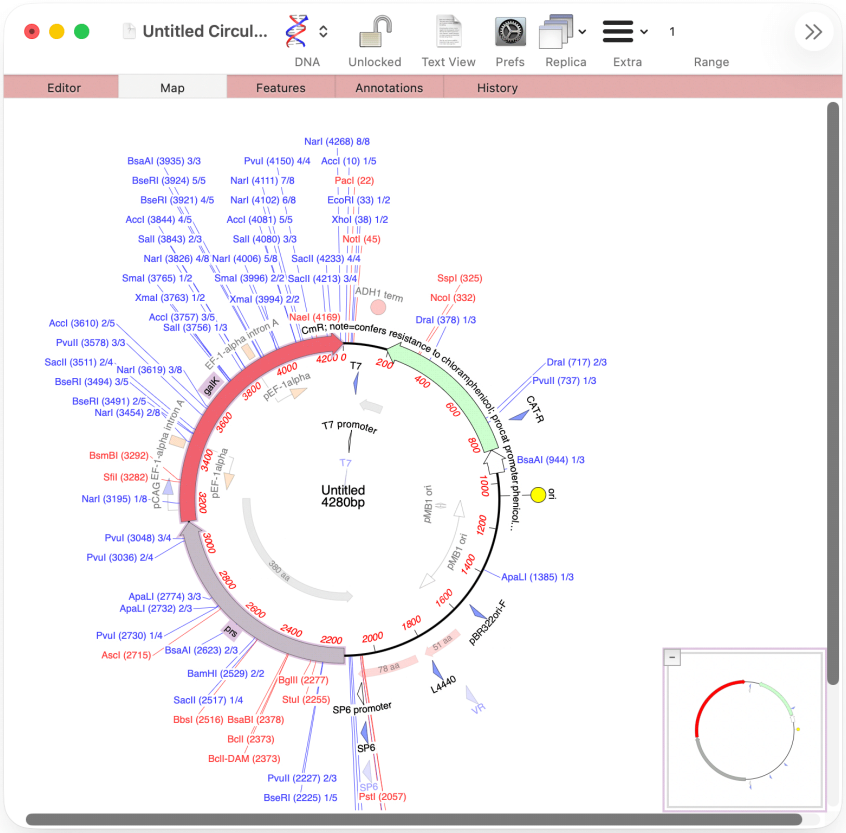
You can print the list via **File | Print** or use the toolbar buttons to save it as a Tab Separated Value (.tsv) file, or add the primers to the current MacVector **Primer Database** file.

Note that MacVector tries to balance the annealing temperatures of all of the primers so that the PCR amplification can be performed in a single multiplex reaction.

Export the Construct

Back in the **Fragments** tab, the small square pane at the lower right indicates that the vector plus the two fragments can indeed be circularized to create a new plasmid given the current set of

primers. If there is a problem, this pane will indicate that the fragments cannot be circularized rather than showing a graphic of the circular molecule. The **Assemble** toolbar button will be active if a circular construct can be generated. Click on **Assemble** and a new window will appear.



This is a standard MacVector annotated single sequence document window, containing the features from each input vector or insert with the exact sequence at each junction. The final construct has no *Bsa* I recognition sequences which is why you can generate this plasmid in a "one-pot" reaction with a very high percentage of the resulting clones having the required sequence. Any recircularizations of the original vector (where the *Bsa* I cleaved vector re-ligates with the original small stuffer fragment) will be recut with *Bsa* I. So, if you run repeated rounds of cut and ligate, finishing with "cut", most of your transformed colonies should contain the correct construct.