

# plasmidcanvas 0.1.0 - Beta install and usage

If you are seeing this, thank you for deciding to take a look at the beta version of plasmidcanvas! This document gives installation and usage instructions for the package as well as full examples of plasmid maps and the code to generate them.

## What is plasmidcanvas?

plasmidcanvas is a work in progress python graphics package designed for producing customised plasmid maps from scratch.

### Current Features as of v0.1.0

- Directional arrows and rectangles to represent features of a plasmid.
- Support for restriction sites
- Support for arbitrary labels
- Support for overlapping features by automatically moving features inwards.
- **Two types of plasmid base pair tick labels**
  - auto - The circle is automatically labelled using the most suitable tick intervals.
  - n\_labels - The circle is given n labels, evenly spaced around the plasmid circle.
- **Two types of feature labels**
  - off-circle - A label is placed outside the plasmid circle, pointing at the base pair / feature of interest.
  - on-circle (curved text) - A label is placed on a feature and curves around the circle with the feature. **Work in progress!**

## Feedback

It would be massively appreciated if you could give me any feedback about this package as you use it. This could be:

- Problems with installing or using the package
- Issues with the generated plasmid maps
- Suggestions for features
- Bugs or inaccuracies

Please provide feedback here: [https://docs.google.com/forms/d/e/1FAIpQLSdVYBA13t9LZIU6dybO8jgwuyIDPQiHB0td0g4EZwGpCSjglQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdVYBA13t9LZIU6dybO8jgwuyIDPQiHB0td0g4EZwGpCSjglQ/viewform?usp=sf_link)

Alternatively, if you want to send me an email please contact [thom.robinson@york.ac.uk](mailto:thom.robinson@york.ac.uk). I am always happy to talk about this project.

## Prerequisites

- Python 3.9 or higher is installed

## Installation

- 1 - Open a command terminal (In windows, press 'windows key' + 'r' then enter 'cmd' to open one)
- 2 - Navigate to the plasmidcanvas folder from the zip (use `cd path/to/the/folder`)
- 3 - Run the following command to install the package: `pip install .`

You now have the package installed! If you want to double check it has installed correctly then run `pip list` to verify `plasmidcanvas` is in the list

## Usage

These steps run through creating a basic, unstyled plasmid map. These examples can be extended using the techniques shown here and in the "Customising your plasmid map" section later on.

### 1 - Import all of ```plasmidcanvas.plasmid``` and ```plasmidcanvas.feature``` into a new Python file

```
from plasmidcanvas.plasmid import *
from plasmidcanvas.feature import *
```

### 2 - Create a new Plasmid object, passing through a name and a number of base pairs.

```
# Creates a plasmid that is 2500 base pairs long and is called "my_plasmid"
plasmid = Plasmid("my_plasmid", "2500")
```

### 3 - Create and add the plasmid's features to the plasmid

At the moment only `RectangleFeature` and `ArrowFeature` can be used to represent multi-pair features. Note that these features will automatically be labelled with their name and their base-pair range.

```
# Creates a rectangle to represent a feature called "some_gene", spanning from bp (basepair) 500 to bp 1000
some_gene = RectangleFeature("some_gene", 500, 1000)
plasmid.add_feature(some_gene)

# Creates a clockwise arrow to represent a feature called "another_gene" spanning from bp 2000 to 2300
another_gene = ArrowFeature("another_gene", 2000, 2300)
plasmid.add_feature(another_gene)

# Creates a counter-clockwise arrow from bp 300 to bp 400 to represent "ori"
ori = ArrowFeature("ori", 300, 400)
plasmid.add_feature(ori)
```

### 4 - Add any restriction sites or additional labels you want.

`RestrictionSite` takes a name and a base pair and formats a label at that base pair location with the text {name} ({basepair}) `SinglePairLabel` works the same, except whatever text it is given will be exactly what is displayed on the label, allowing you to add an arbitrary label.

```
# Creates a restriction site, this will create a label with the text "AbcD (900)" at bp 900
abcd = RestrictionSite("AbcD", 900)
plasmid.add_feature(abcd)

# Creates a label to mark where something might be
label = SinglePairLabel("Some extra label", 1500)
plasmid.add_feature(label)
```

### 5 - Save the plasmid out to a file, giving it a filename. Note that the extension on the filename will determine the filetype. Currently this is only tested for .png and .pdf but any matplotlib supported filetype should work.

```
plasmid.save_to_file("example_plasmid.png")
```

### 6 - Run your script and view the file `example_plasmid.png` It should be in the same directory as your Python script. However, you may notice it looks a little bit **boring**... See the section below on customising you map to avoid this.

Note - If you don't feel like adding in all the details for a plasmid, there is a code example in the Examples section later. Example 1 has the code to generate a uncustomised version of pBR322.

## Customising your plasmid map

Below are some examples of how you can customise your plasmid maps and its features at a fine grained level. In the future there will be support for simpler ways to apply "styles" to your plasmid map / feature types.

### Changing the color of a feature

```
ori = ArrowFeature("ori", 2534, 3122, direction=-1)
ori.set_color("green")
plasmid.add_feature(ori)
```

### Changing the font color or font size of a label or restriction site

This example also applies for RestrictionSite objects.

```
# Creates a label to mark where something might be
label = SinglePairLabel("Some label", 1500)
# Sets the labels font color to red
label.set_font_color("red")
# Set the font size to 10pt
label.set_font_size(10)
plasmid.add_feature(label)
```

### Changing the color or length of a label or restriction site

This example also applies for RestrictionSite objects.

```
# Creates a label to mark where something might be
label = SinglePairLabel("Some label", 1500)
# Scale factor to increase the line length by
label.set_line_length_sf(1.25)
# Set the line color to red
label.set_line_color("red")
plasmid.add_feature(label)
```

### Changing the width of a rectangle feature

Note - The same should be possible for ArrowFeature objects in the future

```
rct = ArrowFeature("rectangle", 2534, 3122)
# Makes the width of the arrow 1.25 times wider than the width of the plasmid circle
rct.set_line_width_scale_factor(1.25)
plasmid.add_feature(ori)
```

### Changing the plasmid line width

The following code can be used to make the plasmid line width wider or thinner. Note that this will increase in line width will be passed down to all features at render time.

```
plasmid = Plasmid("myplasmid", 5000)
# Create a new line width that is 1.25x larger than before
new_line_width = plasmid.get_plasmid_line_width() * 1.25
plasmid.set_plasmid_line_width()
```

## Changing the base pair tick marker style for a Plasmid

There are two types of plasmid base pair tick labels

- auto - (default) The circle is automatically labeled using the most suitable tick intervals.
- n\_labels - The circle is given n labels, evenly spaced around the plasmid circle.

Auto is the default label style, n\_labels can be used as below. If unspecified n=16.

```
plasmid = Plasmid("myplasmid", 5000)
plasmid.set_marker_style("n_labels")
# By default n=16, to change this do:
plasmid.set_number_of_markers(8)
```

## Changing the distance of marker text from the circle

This may lead to some text clipping into labels, but the option is here if you need to change this.

```
plasmid = Plasmid("myplasmid", 5000)
# Sets the markers 1.25x the distance away from the circle when compared to the default
plasmid.set_marker_distance_sf(1.25)
```

## Example 1 - Creating a map of pBR322

The following code shows a concrete example of producing a basic, unstyled map of pBR322

```
# An example showing how to build pBR322 in plasmidcanvas

from plasmidcanvas.plasmid import *
from plasmidcanvas.feature import *

plasmid = Plasmid("pBR322", 4361)

# Adding features
tcr = ArrowFeature("TcR", 86, 1276)
plasmid.add_feature(tcr)

bom = RectangleFeature("bom", 2208, 2348)
plasmid.add_feature(bom)

ori = ArrowFeature("ori", 2534, 3122, direction=-1)
plasmid.add_feature(ori)

ampr = ArrowFeature("ampr", 3293, 4153, direction=-1)
plasmid.add_feature(ampr)

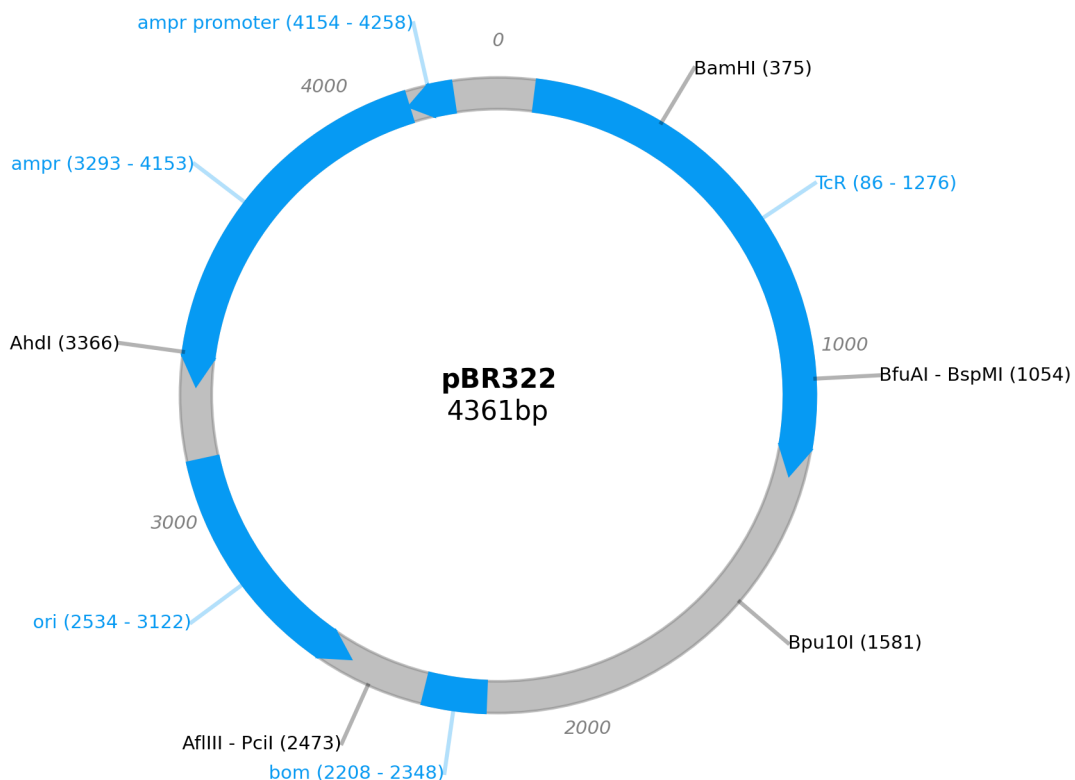
ampr_promoter = ArrowFeature("ampr promoter", 4154, 4258, direction=-1)
plasmid.add_feature(ampr_promoter)

# Add a couple of restriction sites to the plasmid
restriction_site_1 = RestrictionSite("BamHI", 375)
restriction_site_2 = RestrictionSite("BfuAI - BspMI", 1054)
restriction_site_3 = RestrictionSite("Bpu10I", 1581)
restriction_site_4 = RestrictionSite("AflIII - PciI", 2473)
restriction_site_5 = RestrictionSite("AhdI", 3366)
```

```
# Add the sites to the plasmid
plasmid.add_feature(restriction_site_1)
plasmid.add_feature(restriction_site_2)
plasmid.add_feature(restriction_site_3)
plasmid.add_feature(restriction_site_4)
plasmid.add_feature(restriction_site_5)

plasmid.save_to_file("pBR322_basic.png")
```

This produces the following map as a png in your script's directory



## Example 2 - Demonstrating overlapping features on pBR322

This is an example to show how overlapping features look in plasmidcanvas

```
plasmid = Plasmid("pBR322", 4361)

# Adding an arrow
# for pBR322 this is TcR
tcr = ArrowFeature("TcR", 86,1276)
# # # Customise the thickness of the line relative to the thickness of the plasmid circle
# # tcr.set_line_width_scale_factor(1.0)
plasmid.add_feature(tcr)
```

```
# # Add rop protein for pBR322
rop = ArrowFeature("rop", 1915, 2106)
plasmid.add_feature(rop)

# # Add a rectangle, base of mobility for pBR322
bom = RectangleFeature("bom", 2208, 2348)
plasmid.add_feature(bom)

# # Add ori
ori = ArrowFeature("ori", 2534, 3122, -1)
ori.set_color("orange")
plasmid.add_feature(ori)

# # Add ampr - technically this arrow should have a portion segmented for its signal sequence
ampr = ArrowFeature("ampr", 3293, 4153, -1)
ampr.set_color("red")
plasmid.add_feature(ampr)

# # Add ampr promoter as an arrow
ampr_promoter = ArrowFeature("ampr promoter", 4154, 4258, -1)
ampr_promoter.set_color("darkred")
plasmid.add_feature(ampr_promoter)

overlapping = ArrowFeature("overlapping feature", 3500, 4300)
overlapping.set_color("darkblue")
plasmid.add_feature(overlapping)

overlapping = ArrowFeature("overlapping feature2", 3366, 3440)
overlapping.set_color("darkgreen")
plasmid.add_feature(overlapping)

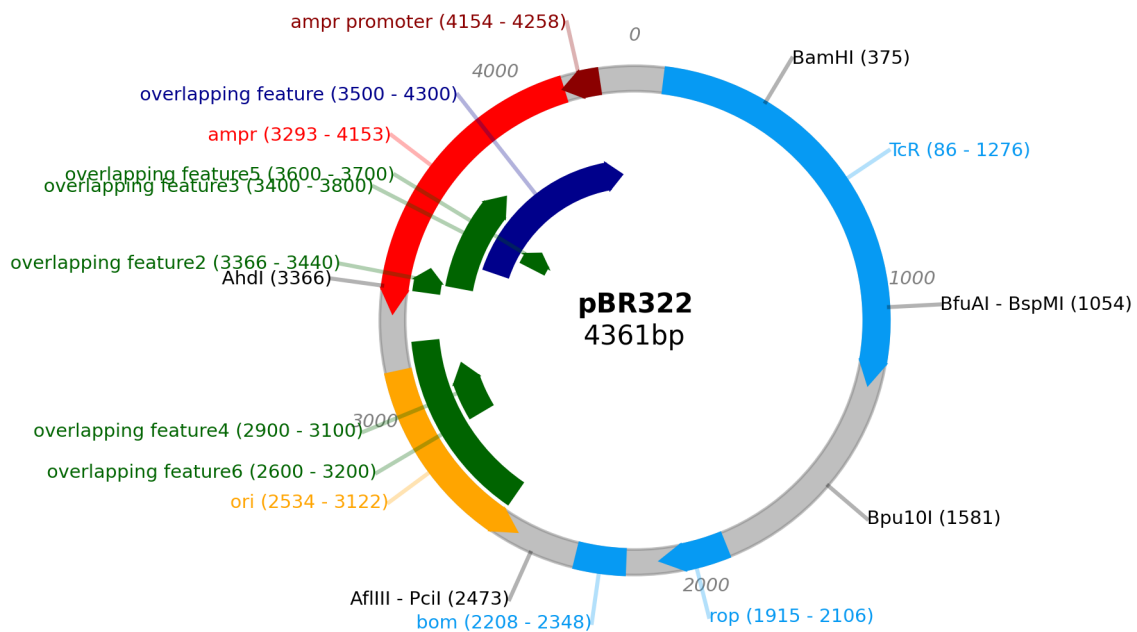
overlapping = ArrowFeature("overlapping feature3", 3400, 3800)
overlapping.set_color("darkgreen")
plasmid.add_feature(overlapping)

overlapping = ArrowFeature("overlapping feature4", 2900, 3100)
overlapping.set_color("darkgreen")
plasmid.add_feature(overlapping)

overlapping = ArrowFeature("overlapping feature5", 3600, 3700)
overlapping.set_color("darkgreen")
plasmid.add_feature(overlapping)

overlapping = RectangleFeature("overlapping feature6", 2600, 3200)
overlapping.set_color("darkgreen")
plasmid.add_feature(overlapping)

plasmid.save_to_file("myplasmid")
```



## Future of the package

### Upcoming Features:

- Full automated api style documentation and extended
- More shapes!
- More labelling styles - curved text, inside the circle labelling, multiple labels per feature
- Base pair ticks
- Extensions to the feature types e.g. allowing you to add Promoter instead of an ArrowFeature
- Plasmid wide styling - i.e applying a style to all elements, sort of like a theme
- Feature wide styling - i.e applying the same style to all of a type of feature

If you want to know any more please contact [thom.robinson@york.ac.uk](mailto:thom.robinson@york.ac.uk)

Remember to fill out the feedback form [https://docs.google.com/forms/d/e/1FAIpQLSdVYBA13t9LZIU6dybO8jgwuyIDPQiHB0td0g4EZwGpCSjglQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdVYBA13t9LZIU6dybO8jgwuyIDPQiHB0td0g4EZwGpCSjglQ/viewform?usp=sf_link)