

---

# **plasmidcanvas**

***Release 1.0.0***

**Thom Robinson**

**Apr 28, 2024**



## CONTENTS:

<b>1</b>	<b>What is plasmidcanvas?</b>	<b>1</b>
<b>2</b>	<b>Bugs, feature requests and contributing</b>	<b>3</b>
<b>3</b>	<b>Prerequisites</b>	<b>5</b>
<b>4</b>	<b>Installation</b>	<b>7</b>
<b>5</b>	<b>Usage</b>	<b>9</b>
<b>6</b>	<b>Tutorial</b>	<b>11</b>
<b>7</b>	<b>Customising your plasmid map</b>	<b>13</b>
7.1	Changing the color of a feature . . . . .	13
7.2	Changing the font color or font size of a label or restriction site . . . . .	13
7.3	Changing the color or length of a label or restriction site . . . . .	13
7.4	Changing the width of a rectangle feature . . . . .	14
7.5	Changing the plasmid line width . . . . .	14
7.6	Changing the base pair tick marker style for a Plasmid . . . . .	14
7.7	Changing the distance of marker text from the circle . . . . .	15
7.8	Using on-circle labelling (curved text) . . . . .	15
7.9	Changing font size for all labels . . . . .	15
<b>8</b>	<b>Example 1 - Creating a map of pBR322</b>	<b>17</b>
<b>9</b>	<b>Example 2 - Demonstrating overlapping features on pBR322</b>	<b>19</b>
<b>10</b>	<b>Example 3 - pBR322 with curved text, mixed labels and more</b>	<b>21</b>
<b>11</b>	<b>API Reference</b>	<b>23</b>
11.1	plasmidcanvas . . . . .	23
	<b>Python Module Index</b>	<b>41</b>
	<b>Index</b>	<b>43</b>



## WHAT IS PLASMIDCANVAS?

plasmidcanvas is a Python graphics package designed for producing customised plasmid maps.

### Top level overview of plasmidcanvas' current features as of v1.0.0

- 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.
- Support for base pair “ticks”.
- **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.
- Plasmids can be saved to a variety of filetypes e.g. png, pdf, ps, eps and svg.



## **BUGS, FEATURE REQUESTS AND CONTRIBUTING**

**## How Can I Contribute?** If you are a user who would like to contribute to the project, please refer to the CONTRIBUTING.md document.

### **### Reporting Bugs**

If you encounter a bug or unexpected behavior, please search the [existing issues](<https://github.com/th0mr/plasmidcanvas/issues>) to see if it has already been reported. If not, please [open a new issue](<https://github.com/th0mr/plasmidcanvas/issues/new>) and provide a clear description of the problem along with steps to reproduce it.

### **### Requesting Features**

If you have a feature request or an enhancement idea, please search the [existing issues](<https://github.com/th0mr/plasmidcanvas/issues>) to see if it has already been requested. If not, please [open a new issue](<https://github.com/th0mr/plasmidcanvas/issues/new>) and describe the feature or enhancement you'd like to see.

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 prompt with admin privileges and run `pip install plasmidcanvas`

2 - 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

This document contains example usage of the package through tutorials, options and example. **However we advise that users take a look at the api documentation for more detail on how methods work and their parameters. See API Reference on the sidebar**



## TUTORIAL

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} ({base-pair}). 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)
```

(continues on next page)

(continued from previous page)

```
# 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 focuses on customising your map to avoid this.

## CUSTOMISING YOUR PLASMID MAP

Below are some examples of how you can customise your plasmid maps and its features at a fine grained level.

### 7.1 Changing the color of a feature

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

### 7.2 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)
```

### 7.3 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)
```

## 7.4 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)
```

## 7.5 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(new_line_width)
```

Or, apply a scale factor to the line width

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

## 7.6 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)
```

## 7.7 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)
```

## 7.8 Using on-circle labelling (curved text)

To swap a label to use on-circle labelling, a new style array must be passed to the feature. If your feature is too small to fit the label on, it won't be placed.

Note - it is possible to have both on-circle and off-circle styles by passing in ["on-circle", "off-circle"]

```
ori = ArrowFeature("ori", 2534, 3122, direction=-1)
ori.set_label_styles(["on-circle"])
plasmid.add_feature(ori)
```

## 7.9 Changing font size for all labels

If you wish to easily change the font size on all labels associated with the Plasmid and its Features, it can be set with `Plasmid.set_label_font_size()`

Note - You can still alter the size of any specific label manually, e.g. `label.set_font_size()` and that won't be overridden by this setting i.e. any manually changed label size won't have the global font size applied to it.

```
plasmid = Plasmid("myplasmid", 5000)
# Accepts a pt value
plasmid.set_label_font_size(5)
```



## 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 Plasmid
from plasmidcanvas.feature import ArrowFeature, RectangleFeature, RestrictionSite

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)

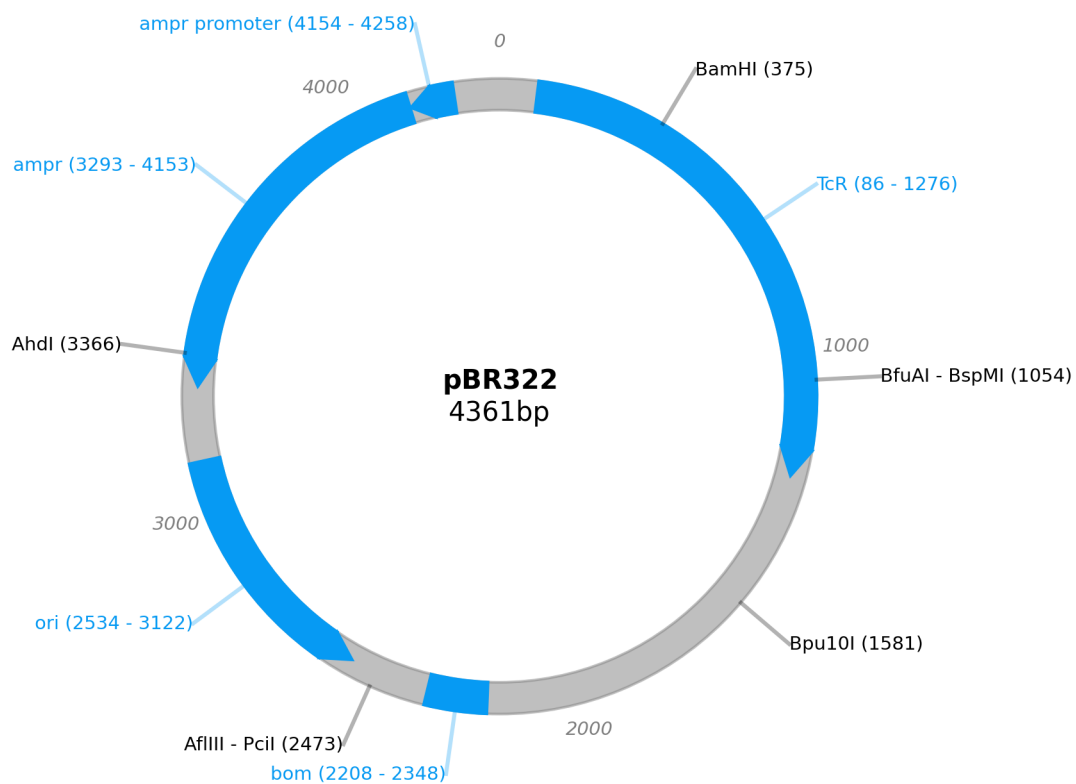
ampr_promoter = ArrowFeature("ampr promoter", 4154, 4258, direction=-1)
plasmid.add_feature(ampr_promoter)

# 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

```
from plasmidcanvas.plasmid import Plasmid
from plasmidcanvas.feature import ArrowFeature, RectangleFeature, RestrictionSite

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)
```

(continues on next page)

(continued from previous page)

```
overlapping.set_color("darkblue")
plasmid.add_feature(overlapping)

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

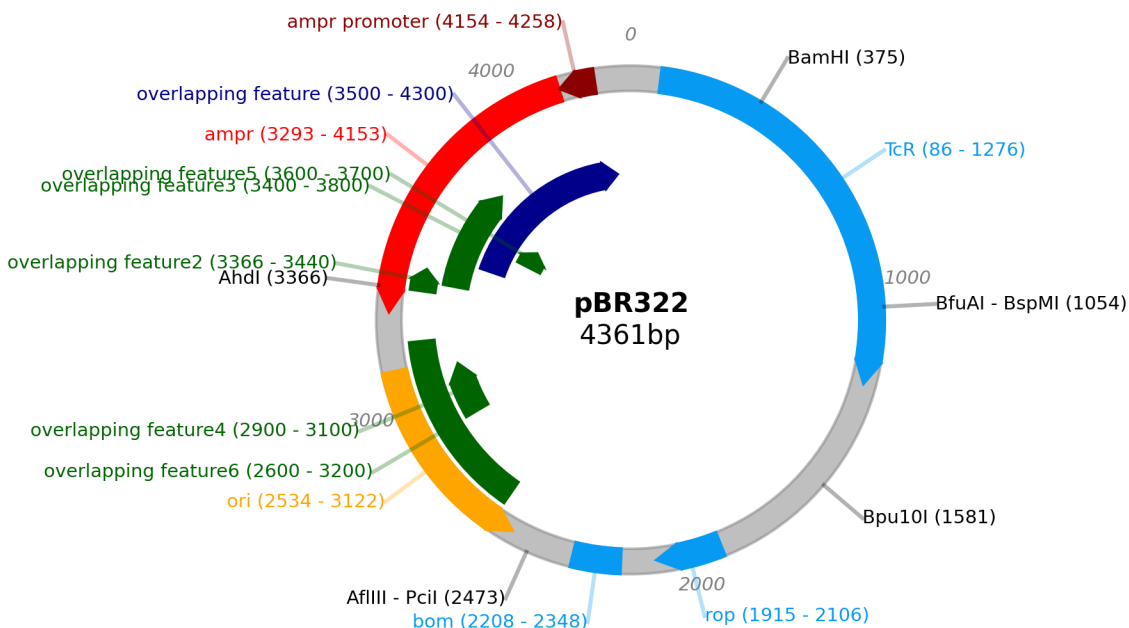
overlapping = ArrowFeature("overlapping feature3", 3400, 3800)
overlapping.set_color("darkgreen")
plasmid.add_feature(overlapping)

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.png")
```



## EXAMPLE 3 - PBR322 WITH CURVED TEXT, MIXED LABELS AND MORE

```
from plasmidcanvas.plasmid import Plasmid
from plasmidcanvas.feature import ArrowFeature, RectangleFeature, RestrictionSite

# Define a plasmid of X base pairs long, with a name
plasmid = Plasmid("pBR322", 4361)
plasmid.set_marker_style("auto")
plasmid.set_feature_label_font_size(7)
plasmid.set_plasmid_line_width_sf(1.25)

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

# Add rop protein for pBR322
rop = ArrowFeature("rop", 1915, 2106)
rop.set_line_width_scale_factor(1.5)
rop.set_color("purple")
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
ampr = ArrowFeature("ampr", 3293, 4153, -1)
ampr.set_color("red")
plasmid.add_feature(ampr)

for feature in plasmid.get_features():
    feature.set_label_styles(["on-circle"])

# # Add ampr promoter as an arrow
ampr_promoter = ArrowFeature("ampr promoter", 4154, 4258, -1)
```

(continues on next page)

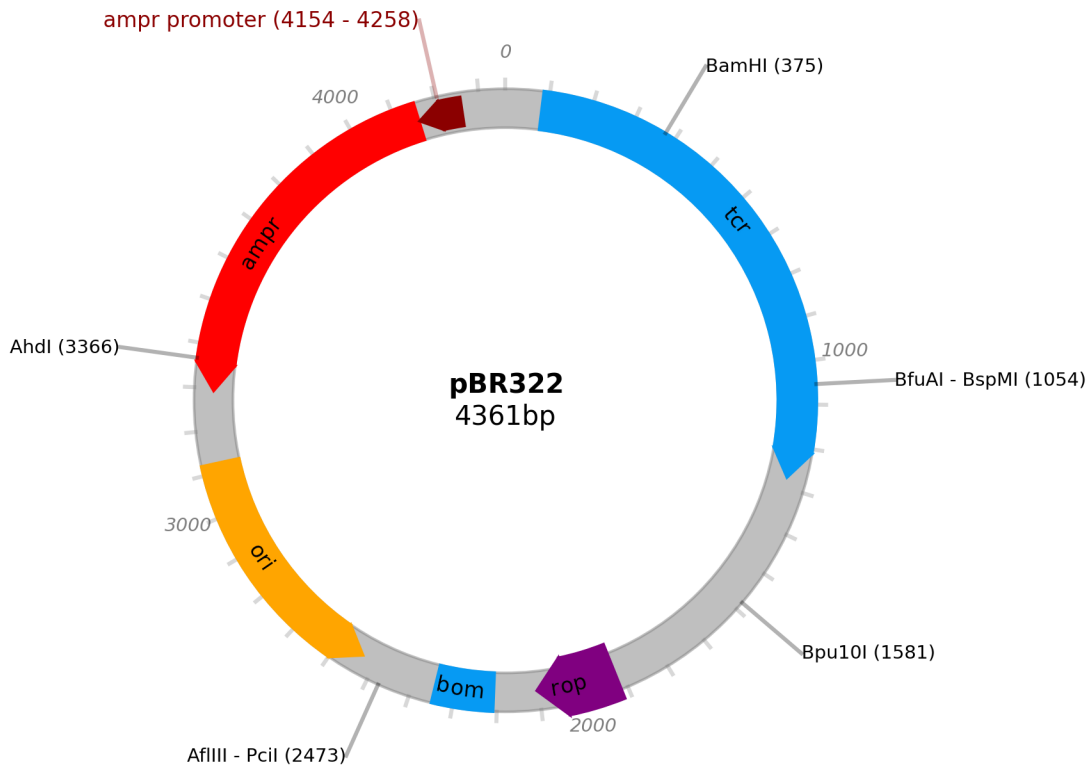
(continued from previous page)

```
ampr_promoter.set_color("darkred")

ampr_promoter.set_line_width_scale_factor(0.75)
plasmid.add_feature(ampr_promoter)

# Add the sites to the plasmid
plasmid.add_feature(RestrictionSite("BamHI", 375))
plasmid.add_feature(RestrictionSite("BfuAI - BspMI", 1054))
plasmid.add_feature(RestrictionSite("Bpu10I", 1581))
plasmid.add_feature(RestrictionSite("AflIII - PciI", 2473))
plasmid.add_feature(RestrictionSite("AhdI", 3366))

# Plot the plasmid
plasmid.save_to_file("myplasmid.png")
```



## API REFERENCE

This page contains auto-generated API reference documentation<sup>1</sup>.

### 11.1 plasmidcanvas

#### 11.1.1 Submodules

`plasmidcanvas.feature`

##### Module Contents

##### Classes

<i>Feature</i>	Abstract base class representing a feature
<i>MultiPairFeature</i>	An abstract Feature type that spans multiple base pairs
<i>SinglePairFeature</i>	An abstract Feature class representing a Feature that relates to only a single base pair
<i>LabelBase</i>	A base class for labels.
<i>SinglePairLabel</i>	A class representing a label associated with a single base pair feature on a plasmid. Inheriting from both SinglePairLabel and LabelBase.
<i>RestrictionSite</i>	A class representing a restriction site on a plasmid as an extension of SinglePairLabel.
<i>CurvedMultiPairLabel</i>	A class representing a curved label associated with multiple base pairs on a plasmid.
<i>RectangleFeature</i>	A class representing a curved rectangular feature associated with multiple base pairs on a plasmid.
<i>DirectionalMultiPairFeature</i>	An abstract class representing a directional feature associated with multiple base pairs on a plasmid.
<i>ArrowFeature</i>	A class representing a curved arrow feature associated with multiple base pairs on a plasmid.

**class** `plasmidcanvas.feature.Feature(name: str)`

Abstract base class representing a feature

Initializes the Feature object.

---

<sup>1</sup> Created with `sphinx-autoapi`

#### Parameters

**name** (*str*) – The name of the feature.

**name:** **str**

**color:** **str**

**get\_name()** → **str**

Get the name of the feature.

#### Returns

The name of the feature.

#### Return type

**str**

**set\_name**(*name: str*)

Set the name of the feature.

#### Parameters

**name** (*str*) – The name to set for the feature.

**get\_color()** → **str**

Get the color of the feature.

#### Returns

The color of the feature.

#### Return type

**str**

**set\_color**(*color: str*) → **None**

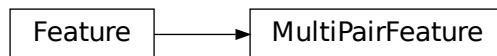
Set the color of the feature.

#### Parameters

**color** (*str*) – The color to set for the feature. Use words e.g “red” or hex values e.g. “#FFFFFF”

**class** `plasmidcanvas.feature.MultiPairFeature`(*name: str, start\_pair: int, end\_pair: int*)

Bases: [Feature](#)



An abstract Feature type that spans multiple base pairs

Initializes the MultiPairFeature object.

#### Parameters

- **name** (*str*) – The name of the feature.
- **start\_pair** (*int*) – The starting base pair of the feature.
- **end\_pair** (*int*) – The ending base pair of the feature.

**start\_pair:** int

**end\_pair:** int

**SUPPORTED\_LABEL\_STYLES** = ['on-circle', 'off-circle']

**label\_style** = ['off-circle']

**length()** → int

Calculates and returns the length of the feature. i.e. end\_pair - start\_pair

**Returns**

The length of the feature.

**Return type**

int

**get\_start\_pair()** → int

Get the starting base pair of the feature.

**Returns**

The starting base pair.

**Return type**

int

**set\_start\_pairs(start\_pair: int)** → None

Set the starting base pair of the feature.

**Parameters**

**start\_pair** – The starting base pair to set.

**get\_end\_pair()** → int

Get the ending base pair of the feature.

**Returns**

The ending base pair.

**Return type**

int

**set\_end\_pairs(end\_pair)** → None

Set the ending base pair of the feature.

**Parameters**

**end\_pair** – The ending base pair to set.

**get\_orbit()** → int

Get the orbit of the feature.

**Returns**

The orbit value of the feature.

**Return type**

int

**set\_orbit(orbit: int)** → None

Set the orbit of the feature. Mostly used internally for overlapping features, but can be used to force features into an orbit. The higher the orbit, the closer to the center of the plasmid the feature is placed. i.e orbit = 0 is the default, orbit = 4 circles four times the width of a feature closer to the center.

**Parameters**

**orbit** (*int*) – The orbit value to set.

**get\_label\_font\_size()** → *int*

Get the font size of the labels associated with the feature.

**Returns**

The font size of the labels.

**Return type**

*int*

**set\_label\_font\_size**(*font\_size: int*) → *None*

Set the font size of the labels associated with the feature.

**Parameters**

**font\_size** (*int*) – The font size to set.

**get\_label\_styles()** → *list[str]*

Get the list of label styles associated with the feature.

**Returns**

**label\_style** – The list of label styles.

**Return type**

*list[str]*

**set\_label\_styles**(*label\_styles: list[str]*)

Set the list of label styles associated with the feature.

**Parameters**

**label\_styles** (*list[str]*) – The list of label styles to set. Currently supported styles

“on-circle” - The label is applied onto the feature using curved text. (The base pair location is not included in this labelling)

“off-circle” - A label is placed off the circle with the name and bp range, connected by a small line

---

**Note:** A plasmid can have both, one or none of the label styles applied at the same time

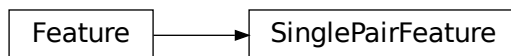
---

**Raises**

**ValueError** – If the given *label\_style* is not in *MultiPairFeature.SUPPORTED\_LABEL\_STYLES* as listed above.

**class** *plasmidcanvas.feature.SinglePairFeature*(*name: str, base\_pair: int*)

Bases: *Feature*



An abstract *Feature* class representing a *Feature* that relates to only a single base pair

Initializes the SinglePairFeature object.

**Parameters**

- **name** (*str*) – The name of the feature.
- **base\_pair** (*int*) – The base pair associated with the feature.

**base\_pair:** *int*

**get\_base\_pair()** → *int*

Get the base pair associated with the feature.

**Returns**

The base pair.

**Return type**

*int*

**set\_base\_pair**(*base\_pair*) → *None*

Set the base pair associated with the feature.

**Parameters**

**base\_pair** – The base pair to set.

**class** plasmidcanvas.feature.**LabelBase**(*name: str*)

A base class for labels.

Initializes the LabelBase object.

**Parameters**

**name** (*str*) – The name of the label.

**DEFAULT\_FONT\_SIZE:** *int*

**DEFAULT\_FONT\_COLOR:** *str* = 'black'

**label\_text:** *str* = 'UntitledLabel'

**font\_color:** *str*

**font\_size:** *int*

**get\_font\_color()** → *str*

Get the color of the label font.

**Returns**

The color of the label font.

**Return type**

*str*

**set\_font\_color**(*font\_color: str*) → *None*

Set the color of the label font.

**Parameters**

**font\_color** (*str*) – The color of the label font.

**set\_label\_text**(*label\_text: str*) → *None*

Set the text of the label.

**Parameters**

**label\_text** (*str*) – The text of the label.

**get\_label\_text()** → str

Get the text of the label.

**Returns**

**label\_text** – The text of the label.

**Return type**

str

**get\_font\_size()** → int

Get the size of the label font.

**Returns**

**font\_size** – The size of the label font.

**Return type**

int

**set\_font\_size(font\_size: int)** → None

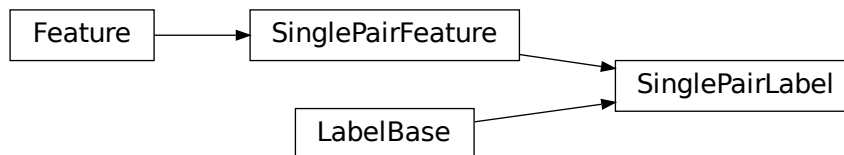
Set the size of the label font.

**Parameters**

**font\_size** (*int*) – The size of the label font.

**class** plasmidcanvas.feature.SinglePairLabel(*name: str, base\_pair: int*)

Bases: [SinglePairFeature](#), [LabelBase](#)



A class representing a label associated with a single base pair feature on a plasmid. Inheriting from both SinglePairLabel and LabelBase.

Initializes the SinglePairLabel object.

**Parameters**

- **name** (*str*) – The name of the label.
- **base\_pair** (*int*) – The base pair associated with the label.

**line\_length\_sf:** float

**line\_color:** str

**get\_line\_color()** → str

Get the color of the line connecting the label to the feature.

**Returns**

The color of the line.

**Return type**

str

**set\_line\_color**(*line\_color: str*) → None

Set the color of the line connecting the label to the feature.

**Parameters**

**line\_color** (*str*) – The color of the line.

**get\_line\_length\_sf**() → float

Get the scale factor for the length of the line connecting the label to the feature.

**Returns**

The scale factor for the line length.

**Return type**

float

**set\_line\_length\_sf**(*line\_length\_sf: float*) → None

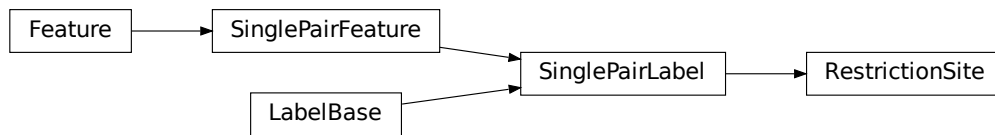
Set the scale factor for the length of the line connecting the label to the feature.

**Parameters**

**line\_length\_sf** (*float*) – The scale factor for the line length.

**class** plasmidcanvas.feature.**RestrictionSite**(*text: str, base\_pair: int*)

Bases: [SinglePairLabel](#)



A class representing a restriction site on a plasmid as an extension of SinglePairLabel.

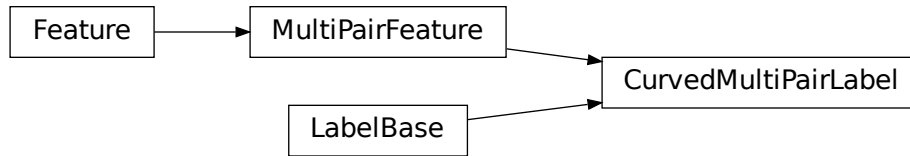
Initializes the RestrictionSite object. Creating a label in the format “<text> (<base\_pair>)”. For more control over the label text, use SinglePairLabel.

**Parameters**

- **text** (*str*) – The text to display on the restriction site. e.g. “BspMI”, “BfuAI, BspMI”, “Multiple Cloning Site”
- **base\_pair** (*int*) – The base pair position of the restriction site.

**class** plasmidcanvas.feature.**CurvedMultiPairLabel**(*name: str, start\_pair: int, end\_pair: int*)

Bases: [MultiPairFeature](#), [LabelBase](#)



A class representing a curved label associated with multiple base pairs on a plasmid.

Initializes the CurvedMultiPairLabel object.

**Parameters**

- **name** (*str*) – The name of the curved label.
- **start\_pair** (*int*) – The starting base pair position of the label.
- **end\_pair** (*int*) – The ending base pair position of the label.

**SUPPORTED\_CURVE\_ALIGNMENTS:** `list[str] = ['bottom', 'top']`

**set\_curve\_alignment**(*curve\_align: str*) → None

Sets the alignment of the curved label. i.e. does the text follow the bottom or top of the curve? (use bottom for curves in the bottom half of the circle and top for the top half)

**Parameters**

- **curve\_align** (*str*) – The alignment of the curved label ('bottom' or 'top').

**get\_curve\_alignment**() → str

Returns the alignment of the curved label.

**Returns**

The alignment of the curved label.

**Return type**

str

**class** plasmidcanvas.feature.RectangleFeature(*name: str, start\_pair: int, end\_pair: int*)

Bases: [MultiPairFeature](#)



A class representing a curved rectangular feature associated with multiple base pairs on a plasmid.

Initializes the MultiPairFeature object.

**Parameters**

- **name** (*str*) – The name of the feature.

- **start\_pair** (*int*) – The starting base pair of the feature.
- **end\_pair** (*int*) – The ending base pair of the feature.

**line\_width\_scale\_factor:** `float = 1`

**get\_line\_width\_scale\_factor()** → `float`

Returns the scale factor for the width of the rectangle.

**Returns**

The scale factor for the width of the rectangle.

**Return type**

`float`

**set\_line\_width\_scale\_factor**(*sf: float*) → `None`

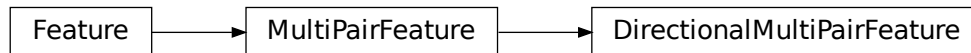
Sets the scale factor for the width of the rectangle.

**Parameters**

**sf** (*float*) – The scale factor for the width of the rectangle.

**class** `plasmidcanvas.feature.DirectionMultiPairFeature`(*name: str, start\_pair: int, end\_pair: int, direction: int = 1*)

Bases: [\*MultiPairFeature\*](#)



An abstract class representing a directional feature associated with multiple base pairs on a plasmid.

Initializes a directional multi-pair feature.

**Parameters**

- **name** (*str*) – The name of the feature.
- **start\_pair** (*int*) – The start base pair.
- **end\_pair** (*int*) – The end base pair.
- **direction** (*int, optional*) – The direction of the feature, either 1 for clockwise or -1 for anti-clockwise (default is 1).

**direction:** `int = 1`

**get\_direction()** → `int`

Returns the direction of the feature.

**Returns**

The direction of the feature. (1 for clockwise or -1 for anti-clockwise)

**Return type**

`int`

**set\_direction**(*direction: int*) → None

Sets the direction of the feature.

**Parameters**

**direction** (*int*) – The direction of the feature, either 1 for clockwise or -1 for anti-clockwise.

**Raises**

**ValueError** – If direction is neither 1 or -1.

**class** plasmidcanvas.feature.**ArrowFeature**(*name: str, start\_pair: int, end\_pair: int, direction: int = 1*)

Bases: *DirectionalMultiPairFeature*



A class representing a curved arrow feature associated with multiple base pairs on a plasmid.

Initializes a directional multi-pair feature.

**Parameters**

- **name** (*str*) – The name of the feature.
- **start\_pair** (*int*) – The start base pair.
- **end\_pair** (*int*) – The end base pair.
- **direction** (*int, optional*) – The direction of the feature, either 1 for clockwise or -1 for anti-clockwise (default is 1).

**line\_width\_scale\_factor:** float = 1

**get\_line\_width\_scale\_factor**() → float

Returns the scale factor for the width of the arrow.

**Returns**

The scale factor for the width of the arrow.

**Return type**

float

**set\_line\_width\_scale\_factor**(*sf: float*) → None

Sets the scale factor for the width of the arrow.

**Parameters**

**sf** (*float*) – The scale factor to set.

plasmidcanvas.plasmid

## Module Contents

### Classes

<i>Plasmid</i>	Circular object representing a plasmid object on which to plot Features onto.
----------------	---

**class** plasmidcanvas.plasmid.**Plasmid**(name: str, base\_pairs: int)  
 Circular object representing a plasmid object on which to plot Features onto.  
 Creates a Plasmid object with a given name and number of base pairs

#### Parameters

- **name** (str) – The name of the plasmid e.g. pBR322
- **base\_pairs** (int) – The number of base pairs in the plasmid

#### Raises

**ValueError** – If the number of base pairs is negative

### Examples

Creating a Plasmid:

```
from plasmidcanvas.plasmid import Plasmid
myplasmid = Plasmid("pBR322", 4361)
```

**SUPPORTED\_MARKER\_STYLES** = ['auto', 'n\_markers', 'none']

**SUPPORTED\_TICK\_STYLES** = ['auto', 'none']

**name:** str

**base\_pairs:** int

**plot()** → matplotlib.figure.Figure

Plots all features added to the Plasmid object onto a matplotlib Figure.

---

**Note:** Unless you are working in an interactive environment, e.g. Jupyter, it is recommended to use `save_to_file()` to view your plasmid instead.

---

#### Returns

**figure** – A matplotlib figure object with the plasmid and its features plotted onto it

#### Return type

Figure

## Example

Obtaining a plasmid Figure:

```
from plasmidcanvas.plasmid import Plasmid
myplasmid = Plasmid("pBR322", 4361)
figure = myplasmid.plot()
proceed to view or work with figure...
```

**save\_to\_file**(*filename: str*) → None

Plots the plasmid by calling `Plasmid.plot()` and saves the figure to an image with a given filename.

### Parameters

**filename** (*str*) –

**filename or path to save the figure to. e.g.**

“myplasmid.png” would save it to the working directory. “build/myplasmid.png” would save the same image inside the “build” folder.

A file extension should be included in the filename. Any matplotlib supported file extension is supported. e.g. png, pdf, ps, eps and svg.

## Examples

To save a plasmid to a png:

```
from plasmidcanvas.plasmid import Plasmid
myplasmid = Plasmid("pBR322", 4361)
myplasmid.savefig("figure.png")
```

**add\_feature**(*feature: plasmidcanvas.feature.Feature*) → None

Adds a feature to the Plasmid object. All features to be plotted must be created and then added to the plasmid this way. See the example below.

### Parameters

**feature** (*Feature*) – A Feature object to add to the plasmid. e.g `RectangleFeature`, `ArrowFeature`, `SinglePairLabel`, `RestrictionSite` etc. See the documentation for `plasmidcanvas.feature` to see the available Feature types and their usage.

### Raises

**ValueError** – If the feature lies out of bounds of the Plasmid’s base pair range

---

**Note:** This does not render the feature, that only happens when `Plasmid.plot()` or `Plasmid.save_to_file()` is ran. Therefore, plasmid wide and feature specific customisations can be made after the feature is added if you wish.

---

## Examples

Adding an Arrow Feature:

```
from plasmidcanvas.plasmid import Plasmid
from plasmidcanvas.feature import ArrowFeature, RectangleFeature, _
↳ RestrictionSite
myplasmid = Plasmid("pBR322", 4361)
myplasmid.savefig("figure.png")

arrow = ArrowFeature(1000, 2000)
# Optional feature customisation here
myplasmid.add_feature(arrow)

# Save plasmid out
myplasmid.save_to_file("figure.png")
```

**get\_features()** → MutableSequence[*plasmidcanvas.feature.Feature*]

Get the features associated with the plasmid.

### Returns

**features** – The features associated with the plasmid.

### Return type

MutableSequence[*Feature*]

**get\_base\_pairs()** → int

Get the number of base pairs in the plasmid.

### Returns

**base\_pairs** – The number of base pairs in the plasmid.

### Return type

int

**set\_base\_pairs(base\_pairs)** → None

Set the number of base pairs in the plasmid.

### Parameters

**base\_pairs** (*int*) – The number of base pairs to set the plasmid to.

**get\_name()** → str

Get the name of the plasmid.

### Returns

**name** – The name of the plasmid.

### Return type

str

**set\_name(name: str)** → None

Set the name of the plasmid.

### Parameters

**name** (*str*) – The name to set for the plasmid.

**get\_color()** → str

Get the color of the plasmid circle.

**Returns**

**color** – The color of the plasmid circle.

**Return type**

str

**set\_color**(*color: str*) → None

Set the color of the plasmid circle.

**Parameters**

**color** (*str*) – The color to set for the plasmid circle. Use words e.g “red” or hex values e.g. “#FFFFFF”

**get\_center**() → tuple[float, float]

Get the center coordinates of the plasmid.

**Returns**

**center** – The (x, y) coordinates of the center of the plasmid.

**Return type**

tuple[float, float]

**set\_center**(*center: tuple[float, float]*) → None

Set the center coordinates of the plasmid.

**Parameters**

**center** (*tuple[float, float]*) – The (x, y) coordinates to set as the center of the plasmid.

**get\_plasmid\_line\_width\_sf**() → float

Get the scale factor for the plasmid circle line width.

**Returns**

**line\_width\_sf** – The scale factor for the plasmid circle line width.

**Return type**

float

**set\_plasmid\_line\_width\_sf**(*line\_width\_sf: float*) → None

Set the scale factor for the plasmid circle line width.

**Parameters**

**line\_width\_sf** (*float*) – The scale factor to set for the plasmid circle line width. e.g. 1.5 makes the plasmid circle 1.5 times as thick as the default

**get\_plasmid\_line\_width**() → float

Get the plasmid line width.

**Returns**

**plasmid\_line\_width** – The plasmid line width.

**Return type**

float

**set\_plasmid\_line\_width**(*plasmid\_line\_width: float*) → None

Set the plasmid line width.

**Parameters**

**plasmid\_line\_width** (*float*) – The plasmid line width to set.

---

**Note:** Either use `set_plasmid_line_width` or `set_plasmid_line_width_sf`. Increasing both values could create a very thick plasmid circle!

---

**get\_marker\_style()** → str

Get the style of markers for the plasmid.

**Returns**

**marker\_style** – The style of markers used for the plasmid.

**Return type**

str

**set\_marker\_style(marker\_style: str)** → None

Set the style of markers for the plasmid.

**Parameters**

**marker\_style** (str) – The style of markers to set for the plasmid. Currently supported styles

“auto” (default) - Automatically apply markers at a reasonable marker interval based on the plasmid size

“n\_markers” - Place n equidistant markers around the circle. The default is 16, unless changed with `Plasmid.set_number_of_markers()`

“none” - No markers added to the circle

**Raises**

**ValueError** – If the given `marker_style` is not in `Plasmid.SUPPORTED_MARKER_STYLES`, as listed above.

**get\_marker\_distance\_sf()** → float

Get the scale factor for the distance between the circle and the marker text.

**Returns**

**marker\_distance\_sf** – The scale factor for the marker distance.

**Return type**

float

**set\_marker\_distance\_sf(marker\_distance\_sf)** → None

Set the scale factor for the distance between the circle and the marker text.

**Parameters**

**marker\_distance\_sf** (float) – The scale factor to set for the marker distance.

## Examples

This value is small by default (1.03). It is advised to alter this by applying a to the existing scale factor to achieve reliable increases.:

```
myplasmid.set_marker_distance(myplasmid.get_marker_distance_sf() * 1.5)
```

**get\_number\_of\_markers()** → int

Get the number of markers on the plasmid. Only used when the plasmid’s marker style is “n\_markers”.

**Returns**

**number\_of\_markers** – The number of markers on the plasmid.

**Return type**

int

**set\_number\_of\_markers**(*number\_of\_markers: int*) → None

Set the number of markers to produce around the plasmid when the plasmid's marker style is "n\_markers".

**Parameters**

**number\_of\_markers** (*int*) – The number of markers to place around the plasmid when using n\_marker style.

**get\_tick\_style**() → str

Get the style of tick placement on the plasmid.

**Returns**

**tick\_style** – The style of ticks used for the plasmid.

**Return type**

str

**set\_tick\_style**(*tick\_style: str*) → None

Set the style of tick placement on the plasmid.

**Parameters**

**tick\_style** (*str*) – The style of ticks to set for the plasmid.

Currently supported:

“auto” (default) - Automatically draw on ticks at a reasonable marker interval based on the plasmid size

“none” - No ticks are drawn around the circle

**Raises**

**ValueError** – If the given tick\_style is not in Plasmid.SUPPORTED\_TICK\_STYLES, as listed above.

**get\_tick\_color**() → str

Get the color of ticks for the plasmid.

**Returns**

**tick\_color** – The color of ticks used for the plasmid.

**Return type**

str

**set\_tick\_color**(*tick\_color: str*) → None

Set the color of ticks for the plasmid.

**Parameters**

**tick\_color** (*str*) – The color of ticks to set for the plasmid. Use words e.g “red” or hex values e.g. “#FFFFFF”

**get\_feature\_label\_font\_size**() → int

Get the override font size of feature labels for the plasmid. This font size will be applied to any labels / features that have not already had their font size changed manually.

**Returns**

**feature\_label\_font\_size** – The font size of feature labels used for the plasmid. Given as a pt value.

**Return type**

int

**set\_feature\_label\_font\_size**(*feature\_label\_font\_size: int*) → None

Set the font size of feature labels for the plasmid. This font size will be applied to any labels / features that have not already had their font size changed manually.

**Parameters**

**feature\_label\_font\_size** (*int*) – The font size of feature labels to set for the plasmid. Given as a pt value.



## PYTHON MODULE INDEX

### p

`plasmidcanvas`, [23](#)

`plasmidcanvas.feature`, [23](#)

`plasmidcanvas.plasmid`, [33](#)



## A

`add_feature()` (*plasmidcanvas.plasmid.Plasmid* method), 34

`ArrowFeature` (class in *plasmidcanvas.feature*), 32

## B

`base_pair` (*plasmidcanvas.feature.SinglePairFeature* attribute), 27

`base_pairs` (*plasmidcanvas.plasmid.Plasmid* attribute), 33

## C

`color` (*plasmidcanvas.feature.Feature* attribute), 24

`CurvedMultiPairLabel` (class in *plasmidcanvas.feature*), 29

## D

`DEFAULT_FONT_COLOR` (*plasmidcanvas.feature.LabelBase* attribute), 27

`DEFAULT_FONT_SIZE` (*plasmidcanvas.feature.LabelBase* attribute), 27

`direction` (*plasmidcanvas.feature.DirectionMultiPairFeature* attribute), 31

`DirectionalMultiPairFeature` (class in *plasmidcanvas.feature*), 31

## E

`end_pair` (*plasmidcanvas.feature.MultiPairFeature* attribute), 25

## F

`Feature` (class in *plasmidcanvas.feature*), 23

`font_color` (*plasmidcanvas.feature.LabelBase* attribute), 27

`font_size` (*plasmidcanvas.feature.LabelBase* attribute), 27

## G

`get_base_pair()` (*plasmidcanvas.feature.SinglePairFeature* method), 27

`get_base_pairs()` (*plasmidcanvas.plasmid.Plasmid* method), 35

`get_center()` (*plasmidcanvas.plasmid.Plasmid* method), 36

`get_color()` (*plasmidcanvas.feature.Feature* method), 24

`get_color()` (*plasmidcanvas.plasmid.Plasmid* method), 35

`get_curve_alignment()` (*plasmidcanvas.feature.CurvedMultiPairLabel* method), 30

`get_direction()` (*plasmidcanvas.feature.DirectionMultiPairFeature* method), 31

`get_end_pair()` (*plasmidcanvas.feature.MultiPairFeature* method), 25

`get_feature_label_font_size()` (*plasmidcanvas.plasmid.Plasmid* method), 38

`get_features()` (*plasmidcanvas.plasmid.Plasmid* method), 35

`get_font_color()` (*plasmidcanvas.feature.LabelBase* method), 27

`get_font_size()` (*plasmidcanvas.feature.LabelBase* method), 28

`get_label_font_size()` (*plasmidcanvas.feature.MultiPairFeature* method), 26

`get_label_styles()` (*plasmidcanvas.feature.MultiPairFeature* method), 26

`get_label_text()` (*plasmidcanvas.feature.LabelBase* method), 27

`get_line_color()` (*plasmidcanvas.feature.SinglePairLabel* method), 28

`get_line_length_sf()` (*plasmidcanvas.feature.SinglePairLabel* method), 29

`get_line_width_scale_factor()` (*plasmidcanvas.feature.ArrowFeature* method), 32

`get_line_width_scale_factor()` (*plasmidcanvas.feature.RectangleFeature* method), 31

`get_marker_distance_sf()` (*plasmidcanvas.plasmid.Plasmid* method), 37

`get_marker_style()` (*plasmidcanvas.plasmid.Plasmid* method), 37

get\_name() (*plasmidcanvas.feature.Feature* method), 24  
 get\_name() (*plasmidcanvas.plasmid.Plasmid* method), 35  
 get\_number\_of\_markers() (*plasmidcanvas.plasmid.Plasmid* method), 37  
 get\_orbit() (*plasmidcanvas.feature.MultiPairFeature* method), 25  
 get\_plasmid\_line\_width() (*plasmidcanvas.plasmid.Plasmid* method), 36  
 get\_plasmid\_line\_width\_sf() (*plasmidcanvas.plasmid.Plasmid* method), 36  
 get\_start\_pair() (*plasmidcanvas.feature.MultiPairFeature* method), 25  
 get\_tick\_color() (*plasmidcanvas.plasmid.Plasmid* method), 38  
 get\_tick\_style() (*plasmidcanvas.plasmid.Plasmid* method), 38

## L

label\_style (*plasmidcanvas.feature.MultiPairFeature* attribute), 25  
 label\_text (*plasmidcanvas.feature.LabelBase* attribute), 27  
 LabelBase (*class in plasmidcanvas.feature*), 27  
 length() (*plasmidcanvas.feature.MultiPairFeature* method), 25  
 line\_color (*plasmidcanvas.feature.SinglePairLabel* attribute), 28  
 line\_length\_sf (*plasmidcanvas.feature.SinglePairLabel* attribute), 28  
 line\_width\_scale\_factor (*plasmidcanvas.feature.ArrowFeature* attribute), 32  
 line\_width\_scale\_factor (*plasmidcanvas.feature.RectangleFeature* attribute), 31

## M

module  
   plasmidcanvas, 23  
   plasmidcanvas.feature, 23  
   plasmidcanvas.plasmid, 33  
 MultiPairFeature (*class in plasmidcanvas.feature*), 24

## N

name (*plasmidcanvas.feature.Feature* attribute), 24  
 name (*plasmidcanvas.plasmid.Plasmid* attribute), 33

## P

Plasmid (*class in plasmidcanvas.plasmid*), 33  
 plasmidcanvas  
   module, 23  
 plasmidcanvas.feature  
   module, 23  
 plasmidcanvas.plasmid

module, 33  
 plot() (*plasmidcanvas.plasmid.Plasmid* method), 33

## R

RectangleFeature (*class in plasmidcanvas.feature*), 30  
 RestrictionSite (*class in plasmidcanvas.feature*), 29

## S

save\_to\_file() (*plasmidcanvas.plasmid.Plasmid* method), 34  
 set\_base\_pair() (*plasmidcanvas.feature.SinglePairFeature* method), 27  
 set\_base\_pairs() (*plasmidcanvas.plasmid.Plasmid* method), 35  
 set\_center() (*plasmidcanvas.plasmid.Plasmid* method), 36  
 set\_color() (*plasmidcanvas.feature.Feature* method), 24  
 set\_color() (*plasmidcanvas.plasmid.Plasmid* method), 36  
 set\_curve\_alignment() (*plasmidcanvas.feature.CurvedMultiPairLabel* method), 30  
 set\_direction() (*plasmidcanvas.feature.DirectionMultiPairFeature* method), 31  
 set\_end\_pairs() (*plasmidcanvas.feature.MultiPairFeature* method), 25  
 set\_feature\_label\_font\_size() (*plasmidcanvas.plasmid.Plasmid* method), 38  
 set\_font\_color() (*plasmidcanvas.feature.LabelBase* method), 27  
 set\_font\_size() (*plasmidcanvas.feature.LabelBase* method), 28  
 set\_label\_font\_size() (*plasmidcanvas.feature.MultiPairFeature* method), 26  
 set\_label\_styles() (*plasmidcanvas.feature.MultiPairFeature* method), 26  
 set\_label\_text() (*plasmidcanvas.feature.LabelBase* method), 27  
 set\_line\_color() (*plasmidcanvas.feature.SinglePairLabel* method), 29  
 set\_line\_length\_sf() (*plasmidcanvas.feature.SinglePairLabel* method), 29  
 set\_line\_width\_scale\_factor() (*plasmidcanvas.feature.ArrowFeature* method), 32  
 set\_line\_width\_scale\_factor() (*plasmidcanvas.feature.RectangleFeature* method), 31  
 set\_marker\_distance\_sf() (*plasmidcanvas.plasmid.Plasmid* method), 37  
 set\_marker\_style() (*plasmidcanvas.plasmid.Plasmid* method), 37  
 set\_name() (*plasmidcanvas.feature.Feature* method), 24

[set\\_name\(\)](#) (*plasmidcanvas.plasmid.Plasmid method*),  
[35](#)  
[set\\_number\\_of\\_markers\(\)](#) (*plasmidcan-  
vas.plasmid.Plasmid method*), [38](#)  
[set\\_orbit\(\)](#) (*plasmidcanvas.feature.MultiPairFeature  
method*), [25](#)  
[set\\_plasmid\\_line\\_width\(\)](#) (*plasmidcan-  
vas.plasmid.Plasmid method*), [36](#)  
[set\\_plasmid\\_line\\_width\\_sf\(\)](#) (*plasmidcan-  
vas.plasmid.Plasmid method*), [36](#)  
[set\\_start\\_pairs\(\)](#) (*plasmidcan-  
vas.feature.MultiPairFeature method*), [25](#)  
[set\\_tick\\_color\(\)](#) (*plasmidcanvas.plasmid.Plasmid  
method*), [38](#)  
[set\\_tick\\_style\(\)](#) (*plasmidcanvas.plasmid.Plasmid  
method*), [38](#)  
[SinglePairFeature](#) (*class in plasmidcanvas.feature*),  
[26](#)  
[SinglePairLabel](#) (*class in plasmidcanvas.feature*), [28](#)  
[start\\_pair](#) (*plasmidcanvas.feature.MultiPairFeature  
attribute*), [24](#)  
[SUPPORTED\\_CURVE\\_ALIGNMENTS](#) (*plasmidcan-  
vas.feature.CurvedMultiPairLabel attribute*),  
[30](#)  
[SUPPORTED\\_LABEL\\_STYLES](#) (*plasmidcan-  
vas.feature.MultiPairFeature attribute*), [25](#)  
[SUPPORTED\\_MARKER\\_STYLES](#) (*plasmidcan-  
vas.plasmid.Plasmid attribute*), [33](#)  
[SUPPORTED\\_TICK\\_STYLES](#) (*plasmidcan-  
vas.plasmid.Plasmid attribute*), [33](#)