

Typical uses of the *CyGnusPlotter*

Author: KS (kzyskgch@gmail.com)

This file describes typical uses of CyGnusPlotter. *APOE* including *Alu* repeats was employed as an example gene, based on a paper of Paik et al. [1]. For using CyGnusPlotter in the character user interface, the brief reference manual and the Component subclasses reference were prepared under *documents* folder. The *simple mode* use is shown in the main text.

Definition of *APOE* structure:

The pairwise sequence of the referenced paper was fetched from the URL below (S2 File):

http://rest.ensembl.org/sequence/id/ENST00000252486?content-type=text/plain;expand_5prime=812;expand_3prime=632

This sequence includes 812 bp 5' flanking sequence and 632 bp 3' flanking sequence. According to Paik et al, the TSS is 44 bp downstream than the current annotation. Therefore, TSSs of the sequence are 813 or 857, when employing the current annotation or the referenced paper.

Ranges of exons and *Alu* sequences are shown below according to the referenced paper:

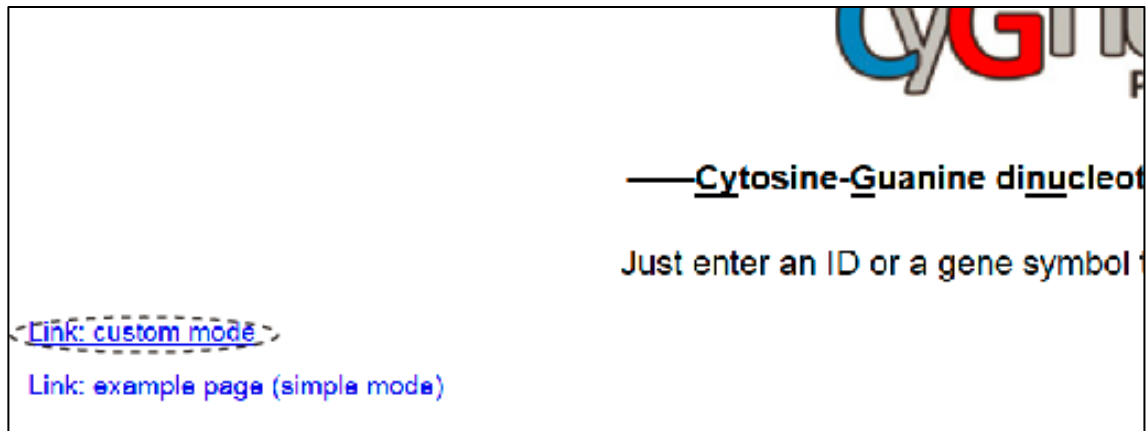
Exon1:	857	-	900
Exon2:	1661	-	1726
Exon3:	2819	-	3011
Exon4:	3594	-	4453
<i>Alu</i> I (leftward):	152	-	441
<i>Alu</i> II (leftward):	1917	-	2240
<i>Alu</i> III (rightward):	2375	-	2667
<i>Alu</i> IV (leftward):	4584	-	4863

Reference

1. Paik YK, Chang DJ, Reardon CA, Davies GE, Mahley RW, Taylor JM. Nucleotide sequence and structure of the human apolipoprotein E gene. PNAS. 1985;82: 3445–3449.

(1) The *custom mode* of the graphical user interface

Step1:



After opening CyGnusPlotterGUI (ref. Results and discussion in the main text), jump to the *custom mode* by clicking a link on left side.

Step2:

The screenshot shows the 'Custom mode' interface of the CyGnusPlotterGUI. At the top right is the logo 'CyGnus Plotter'. Below it, the text '—Cytosine-Guanine dinucleotide sites Plotter—' is displayed. Underneath, it says 'Custom mode: You can get a diagram from a sequence.' On the left side, there is a link 'Link: simple mode'. In the center, there is a 'Title' field with the text 'S1-1'. Below the title field, there is a checkbox labeled 'Display the title' which is checked. Below the checkbox, there is a text box that says 'This option enables the title to be displayed. Users can change the location of the displayed title.' Below the text box, there are three radio buttons: 'left' (which is selected), 'center', and 'right'. Below the radio buttons, there is a 'Sequence' label. Below the label, there is a text area containing a long DNA sequence. At the bottom left, there is a 'submit' button.

Sequence

```
GTCCCGCCCCAGCCGTCCTCTTGGGGTGGACCTAGTTAATAAGATTC  
ACCAAGTTTCACGCATCTGCTGGCCCTCCCCCTGTGATTTCCCTCTAAGCCCC  
AGCCTCAGTTTCTCTTTCTGCCCACATACTGGCCACACAATTCTCAGCCCC  
CTCCTCTCCATCTGTGCTGTGTGTATCTTCTCTCTGCCCTTTTCTTTT  
TTTTAGACGGAGTCTGGCTCTGTACCCAGGCTAGAGTGCAGTGGCAGCAT  
CTTGGCTCACTGCAACCTCTGCCTCTTGGGTCAAGCGATTCTGCTGCCTC  
AGTAGCTGGGATTACAGGCTCACACCACCACACCCGGCTAATTTTGTATT  
TTTAGTAGAGACGAGCTTTCACCATGTTGGCCAGGCAGGTCTCAAACCTCT  
GACCAAGTGATCCACCCGCGGCTCCCAAAGTGCTGAGATTACAGGCTG  
AGCCACCATGCCCGGCTCTGCCCTCTTTCTTTTAGGGGGCAGGGAAA  
GGTCTCACCCCTGTACCCGCCATCACAGCTCACTGCAGCCTCCACCTCCTG  
GACTCAAGTGATAAGTGATCCTCCCGCTCAGCCTTCCAGTAGCTGAGAC  
TACAGGCGCATACCACTAGGATTAATTTGGGGGGGGGGTGGTGTGTGTGG  
AGATGGGGTCTGGCTTTGTGGCCAGGCTGATGTG
```

Enter a title string and a sequence.

Step 3:

Box option

Boxes

Enter [box name], [start position], [end position] to each row.

Assumed purpose of this component class is indicating exons.

1, 857, 900
2, 1661, 1726
3, 2819, 3011
4, 3594, 4453

Arrow option

Arrows

Enter [arrow name], [start position], [end position] to each row.

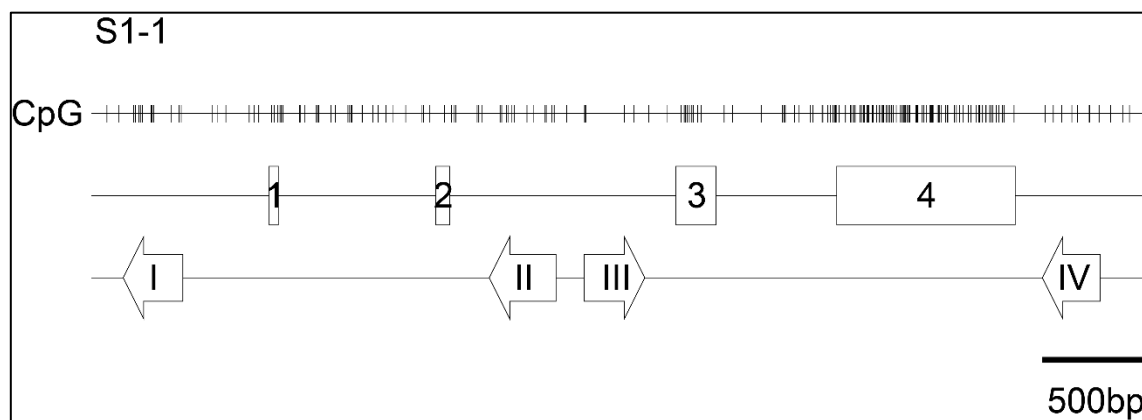
Arrow is similar to Box. However, arrows indicate left or right depending on the input.

If positions inputted larger to smaller, it indicates rightward, and if inputted smaller to larger it indicates leftward.

I, 441, 152
II, 2240, 1917
III, 2375, 2667
IV, 4863, 4584

Enter exon and *Alu* ranges. Arrow directions depend on start and end positions. Probe components can be added in the same way. Click *submit* button and the eps file will be stored.

Output:



(2) The character user interface #1

Description

Make diagram from the sequence. To use script below, *<CyGnusPlotter path>*, *<APOE file>* and *<output path>* should be replaced with strings specifying paths according to a user's system. The sequence is assumed stored in the path a pecified by *<APOE file>* (S2 File).

Python script (Also described in S3 File as a text file.)

```
import os

os.chdir(<CyGnusPlotter path>)

from cyglib import cyginput

#Reading DNA sequence
with open(<APOE file>, 'rb') as apoe:
    seq = apoe.read()

#The 5' flanking sequence length according to Paik et al.
exp5_paik = 856

#Construct a new CyGInput instance
my_input = cyginput.new_input(seq=seq)

#Request the title and scale bar
my_input.accepts_request(component_type='Title',
                        rack_order=my_input.top,
                        name='S1-2',
                        option='left')
my_input.accepts_request(component_type='ScaleBar',
                        rack_order=my_input.bottom,
                        name='bp',
                        end=300,
                        option='right')
```

```

#Request CpG distribution
my_input.accepts_request(component_type='Site',
                        rack_order=2, pattern='cg',
                        position_on_pattern=0,
                        name='CpG', with_shaft=True)

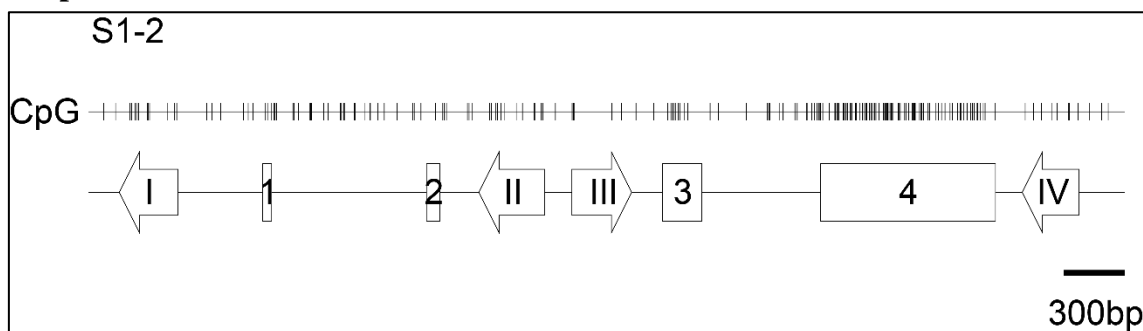
#Request boxes of exons
exons = [('1', 1+exp5_paik, 44+exp5_paik), ('2', 805+exp5_paik, 870+exp5_paik),
        ('3', 1963+exp5_paik, 2155+exp5_paik), ('4', 2738+exp5_paik, 3597+exp5_paik)]
for i in range(len(exons)):
    exon = exons[i]
    my_input.accepts_request(component_type='Box', rack_order=1, name=exon[0],
                            start=exon[1], end=exon[2], with_shaft=i==0)

#Request arrows of Alus
alus = [('I', -415+exp5_paik, -704+exp5_paik), ('II', 1384+exp5_paik, 1061+exp5_paik),
        ('III', 1519+exp5_paik, 1811+exp5_paik), ('IV', 4007+exp5_paik, 3728+exp5_paik)]
for i in range(len(alus)):
    alu = alus[i]
    my_input.accepts_request(component_type='Arrow', rack_order=1, name=alu[0],
                            start=alu[1], end=alu[2])

my_input.scales_layout (360)
my_input.makes_eps_file(<output path>)

```

Output:



(3) The character user interface #2

Description

Make a diagram using an Ensembl ID. By using this feature, users can indicate Infinium probes and SNPs. To use script below, *<CyGnusPlotter path>* and *<output path>* should be replaced with strings specifying paths according to a user's system.

Python script (Also described in S4 File as a text file.)

```
import os

os.chdir(<CyGnusPlotter path>)

from cyglib import cyginput

#Construct a CyGInput instance

gene = 'apoe'

exp5 = 812

exp3 = 632

my_input = cyginput.import_ensembl(gene, exp5, exp3, exon_num='inner')

#5' flanking sequence length according to Paik et al.

exp5_paik = exp5 + 44

#Request the title scale bar, and Infinium probes

my_input.accepts_request(component_type='Title',

                        rack_order=my_input.top,

                        name='S1-3',

                        option='left')

my_input.accepts_request(component_type='ScaleBar',

                        rack_order=my_input.bottom,

                        name='bp', end=300,

                        option='right')

my_input.accepts_request('InfMet', my_input.upper, name='all',

                        option='EPIC', with_shaft=None)
```

```

#Request CpG distribution
my_input.accepts_request(component_type='Site',
                        rack_order=2, pattern='cg',
                        position_on_pattern=0,
                        name='CpG', with_shaft=True)

#Request SNP
my_input.accepts_request('SNP', rack_order=1, name= ['rs7412', 'rs429358'], with_shaft=None)

#Recall request for exon boxes
my_input.recalls_reservation()

#Request arrows of Alus
alus = [('I', -415+exp5_paik, -704+exp5_paik), ('II', 1384+exp5_paik, 1061+exp5_paik),
        ('III', 1519+exp5_paik, 1811+exp5_paik), ('IV', 4007+exp5_paik, 3728+exp5_paik)]
for i in range(len(alus)):
    alu = alus[i]
    my_input.accepts_request(component_type='Arrow', rack_order=1, name=alu[0],
                            start=alu[1], end=alu[2])

my_input.scales_layout (360)
my_input.makes_eps_file(<output path>)

```

Output:

