KBase Analysis Notebook

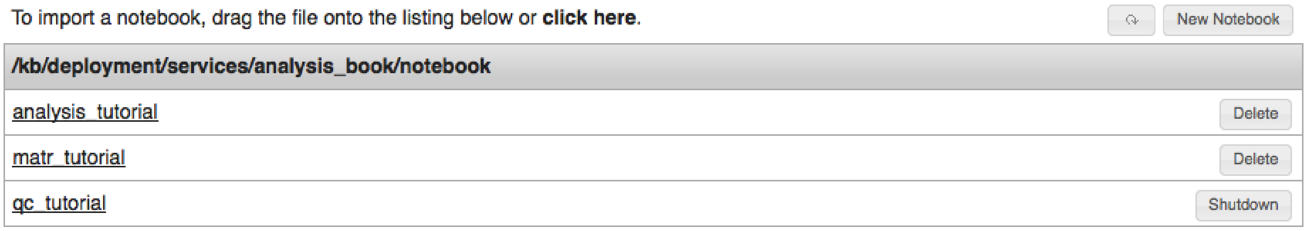
A ipython notebook with bindings to kbase APIs and built in analysis and QC tools

Description

ipython notebook server http://ipython.org/ipython-doc/rel-0.13/overview.html. This is a web based python shell interface with ability to use other shells (bash, perl, R). Allows execution of code blocks (called cells) and display of the results as plain text or images or markup (html / javascript).

Basic Usage

Launch Page



To import a notebook, drag the file onto the listing below or **click here**.

/kb/deployment/services/analysis_book/notebook

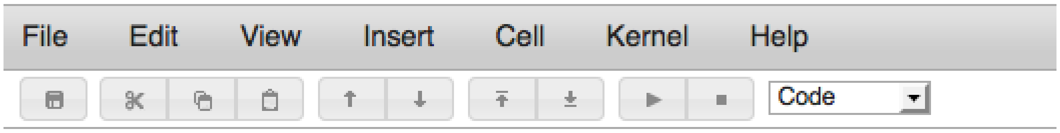
analysis_tutorial
matr_tutorial
qc_tutorial

Here we have the login / launch page which lists all the users available notebooks. A notebook is a python data structure in plain text that saves all the history of commands and output that occurred in notebook session.The user can do the following:

1. create new notebook
2. launch existing notebook
3. delete existing notebook
4. upload notebook

Notebook Page

Composed of 2 parts.
1. A top control bar with buttons / drop-down menus



2. An interactive notebook space below, composed of ‘cells’.

```
In [ ]:
```

Notebook Usage

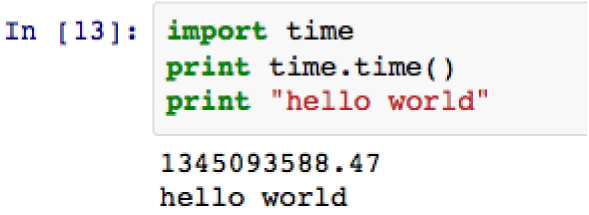
Work within the notebook is done by entering commands into a cell, this can be done with one or more lines. To execute the commands in the cell use shift-return. Cells are numbered by execution order and may be re-ran in any order. The results (if any) are displayed in the space below that cell.

Results may be:

1. plain text
2. JSON or serialized python objects
3. html / markup
4. javascript / svg
5. math equations / latex
6. images: png / jpg / pdf / etc.

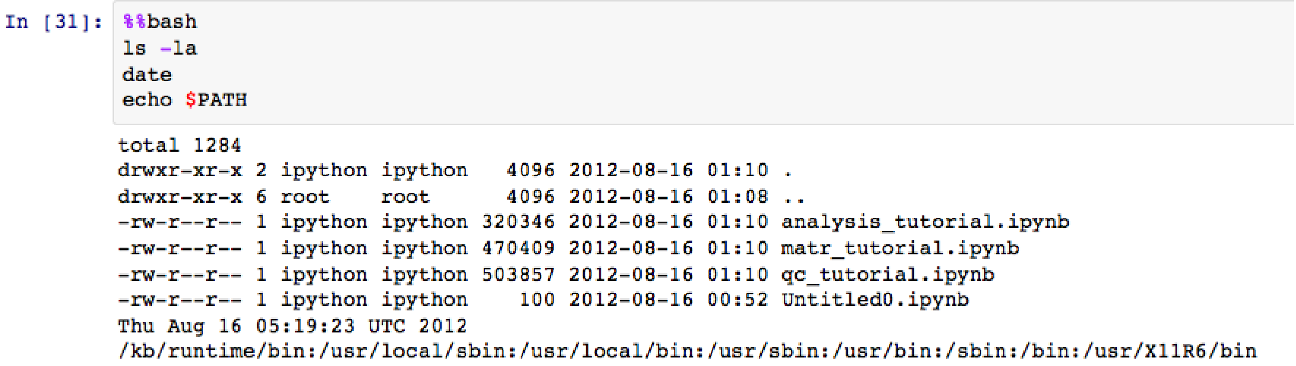
Cell interpreters:

command   cell interproter
1. default   python



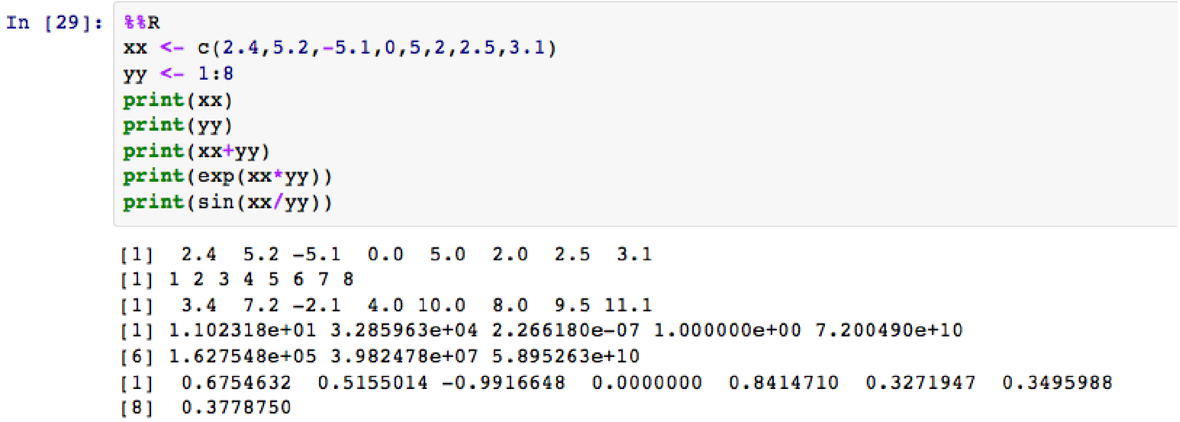
```
In [13]: import time
         print time.time()
         print "hello world"

         1345093588.47
         hello world
```

2. !   system shell (single line only)
3. %%!   system shell
4. %%bash   bash



```
In [31]: %%bash
         ls -la
         date
         echo $PATH

total 1284
drwxr-xr-x 2 ipython ipython   4096 2012-08-16 01:10 .
drwxr-xr-x 6 root    root      4096 2012-08-16 01:08 ..
-rw-r--r-- 1 ipython ipython 320346 2012-08-16 01:10 analysis_tutorial.ipynb
-rw-r--r-- 1 ipython ipython 470409 2012-08-16 01:10 matr_tutorial.ipynb
-rw-r--r-- 1 ipython ipython 503857 2012-08-16 01:10 qc_tutorial.ipynb
-rw-r--r-- 1 ipython ipython    100 2012-08-16 00:52 Untitled0.ipynb
Thu Aug 16 05:19:23 UTC 2012
/kb/runtime/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/X11R6/bin
```

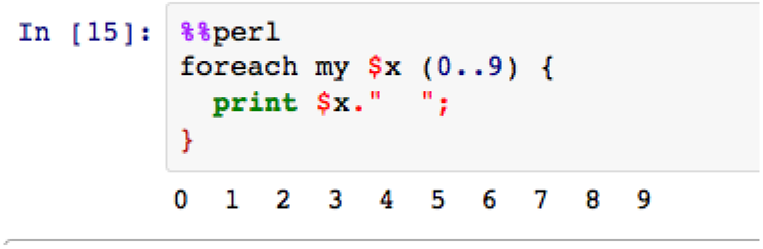
5. %%R   R



```
In [29]: %%R
         xx <- c(2.4,5.2,-5.1,0,5,2,2.5,3.1)
         yy <- 1:8
         print(xx)
         print(yy)
         print(xx+yy)
         print(exp(xx*yy))
         print(sin(xx/yy))
```

```
[1]  2.4  5.2 -5.1  0.0  5.0  2.0  2.5  3.1
[1] 1 2 3 4 5 6 7 8
[1]  3.4  7.2 -2.1  4.0 10.0  8.0  9.5 11.1
[1] 1.102318e+01 3.285963e+04 2.266180e-07 1.000000e+00 7.200490e+10
[6] 1.627548e+05 3.982478e+07 5.895263e+10
[1]  0.6754632  0.5155014 -0.9916648  0.0000000  0.8414710  0.3271947  0.3495988
[8]  0.3778750
```

6. %%perl perl



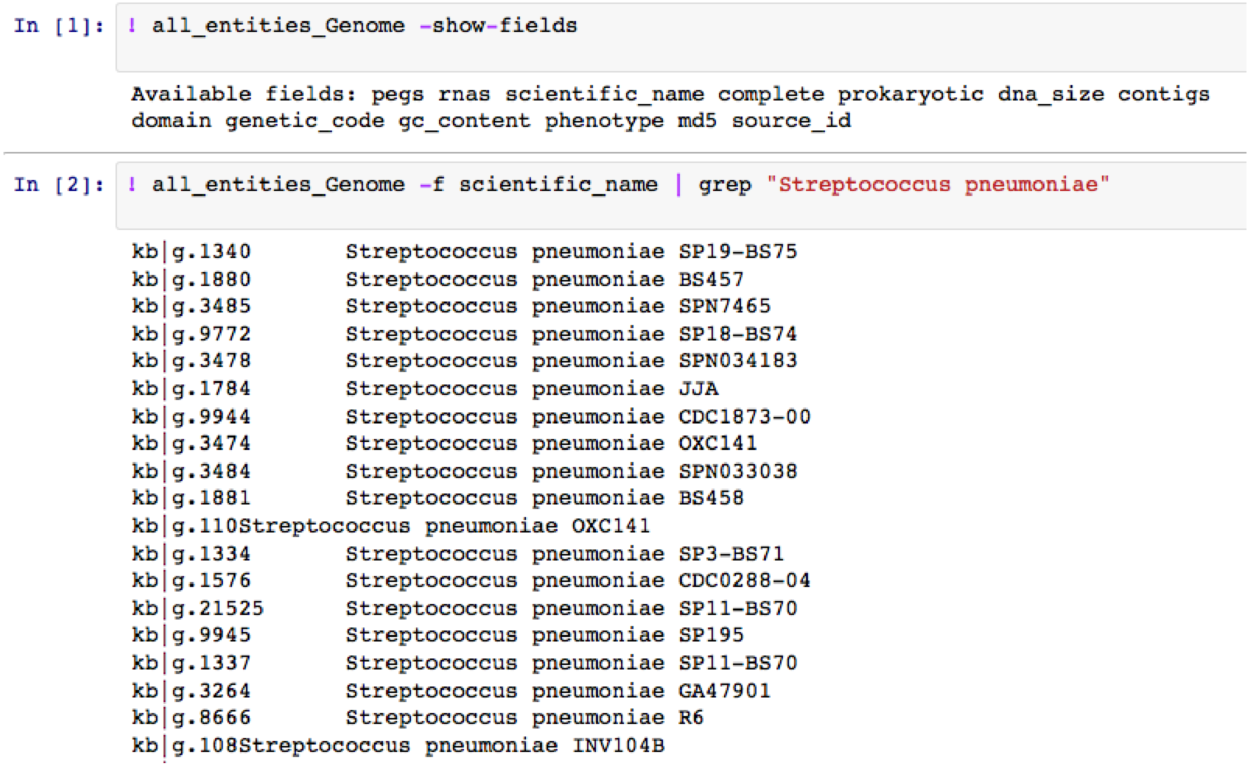
```
In [15]: %%perl
         foreach my $x (0..9) {
           print $x."   ";
         }
```

```
0   1   2   3   4   5   6   7   8   9
```

7. %%ruby ruby
8. %%file X output sent to file X

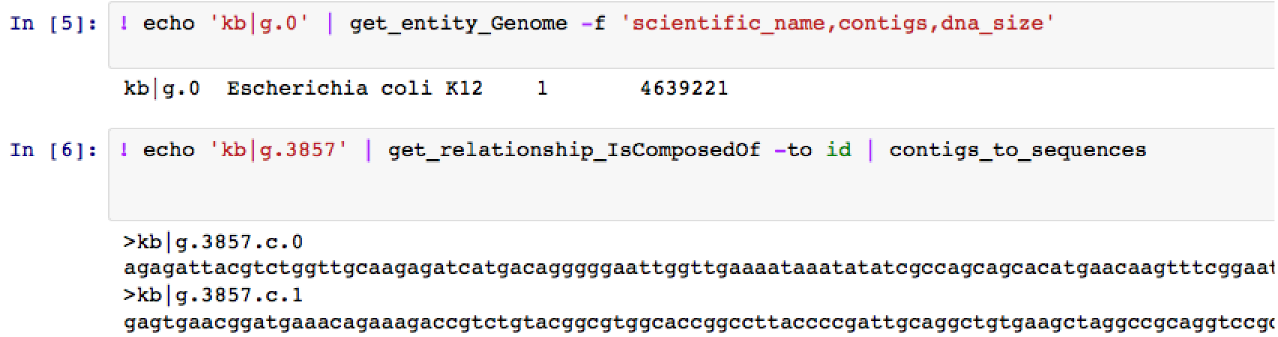
API Usage

1. CDM api



```
In [1]: ! all_entities_Genome -show-fields

        Available fields: pegs rnas scientific_name complete prokaryotic dna_size contigs
        domain genetic_code gc_content phenotype md5 source_id
```

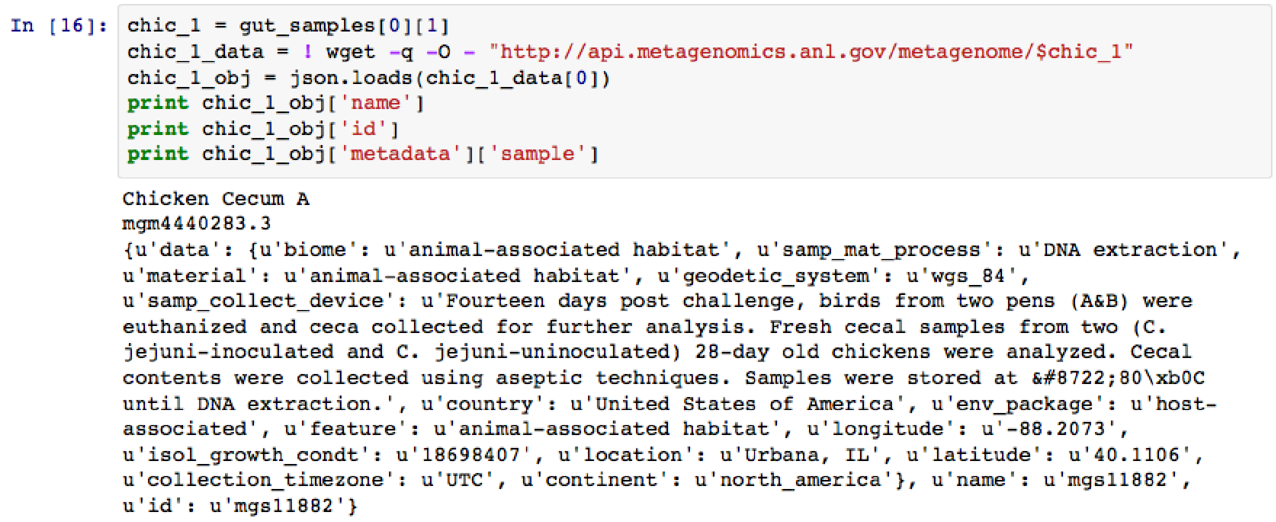
```
In [2]: ! all_entities_Genome -f scientific_name | grep "Streptococcus pneumoniae"

        kb|g.1340       Streptococcus pneumoniae SP19-BS75
        kb|g.1880       Streptococcus pneumoniae BS457
        kb|g.3485       Streptococcus pneumoniae SPN7465
        kb|g.9772       Streptococcus pneumoniae SP18-BS74
        kb|g.3478       Streptococcus pneumoniae SPN034183
        kb|g.1784       Streptococcus pneumoniae JJA
        kb|g.9944       Streptococcus pneumoniae CDC1873-00
        kb|g.3474       Streptococcus pneumoniae OXC141
        kb|g.3484       Streptococcus pneumoniae SPN033038
        kb|g.1881       Streptococcus pneumoniae BS458
        kb|g.110Streptococcus pneumoniae OXC141
        kb|g.1334       Streptococcus pneumoniae SP3-BS71
        kb|g.1576       Streptococcus pneumoniae CDC0288-04
        kb|g.21525      Streptococcus pneumoniae SP11-BS70
        kb|g.9945       Streptococcus pneumoniae SP195
        kb|g.1337       Streptococcus pneumoniae SP11-BS70
        kb|g.3264       Streptococcus pneumoniae GA47901
        kb|g.8666       Streptococcus pneumoniae R6
        kb|g.108Streptococcus pneumoniae INV104B
```



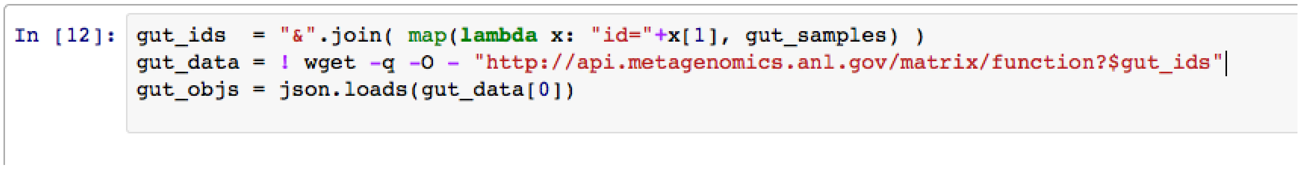
```
In [5]: ! echo 'kb|g.0' | get_entity_Genome -f 'scientific_name,contigs,dna_size'

        kb|g.0  Escherichia coli K12    1       4639221
```

```
In [6]: ! echo 'kb|g.3857' | get_relationship_IsComposedOf -to id | contigs_to_sequences

        >kb|g.3857.c.0
        agagattacgtctggttgcaagagatcatgacaggggaattggttgaaaataaatatatcgccagcagcacatgaacaagtttcggaat
        >kb|g.3857.c.1
        gagtgaacggatgaaacagaaagaccgtctgtacggcgtggcaccggccttaccccgattgcaggctgtgaagctaggccgcaggtccgc
```

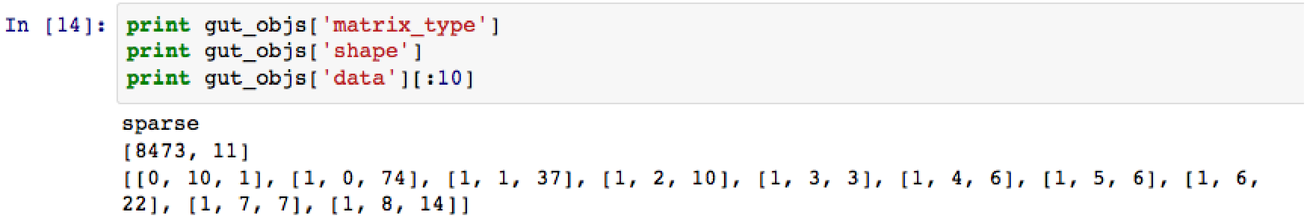
2. REST api



```
In [16]: chic_1 = gut_samples[0][1]
         chic_1_data = ! wget -q -O - "http://api.metagenomics.anl.gov/metagenome/$chic_1"
         chic_1_obj = json.loads(chic_1_data[0])
         print chic_1_obj['name']
         print chic_1_obj['id']
         print chic_1_obj['metadata']['sample']

         Chicken Cecum A
         mgm4440283.3
         {u'data': {u'biome': u'animal-associated habitat', u'samp_mat_process': u'DNA extraction',
         u'material': u'animal-associated habitat', u'geodetic_system': u'wgs_84',
         u'samp_collect_device': u'Fourteen days post challenge, birds from two pens (A&B) were
         euthanized and ceca collected for further analysis. Fresh cecal samples from two (C.
         jejuni-inoculated and C. jejuni-uninoculated) 28-day old chickens were analyzed. Cecal
         contents were collected using aseptic techniques. Samples were stored at &#8722;80\xb0C
         until DNA extraction.', u'country': u'United States of America', u'env_package': u'host-
         associated', u'feature': u'animal-associated habitat', u'longitude': u'-88.2073',
         u'isol_growth_condt': u'18698407', u'location': u'Urbana, IL', u'latitude': u'40.1106',
         u'collection_timezone': u'UTC', u'continent': u'north_america'}, u'name': u'mgs11882',
         u'id': u'mgs11882'}
```



```
In [12]: gut_ids  = "&".join( map(lambda x: "id="+x[1], gut_samples) )
         gut_data = ! wget -q -O - "http://api.metagenomics.anl.gov/matrix/function?$gut_ids"
         gut_objs = json.loads(gut_data[0])
```



```
In [14]: print gut_objs['matrix_type']
         print gut_objs['shape']
         print gut_objs['data'][:10]

         sparse
         [8473, 11]
         [[0, 10, 1], [1, 0, 74], [1, 1, 37], [1, 2, 10], [1, 3, 3], [1, 4, 6], [1, 5, 6], [1, 6,
         22], [1, 7, 7], [1, 8, 14]]
```

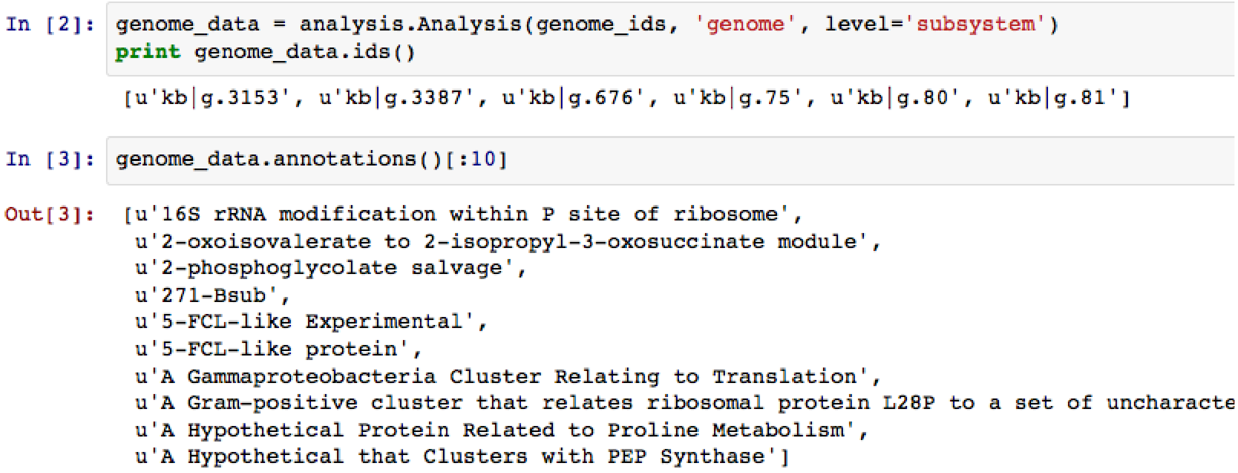
Analysis Usage

workflow:

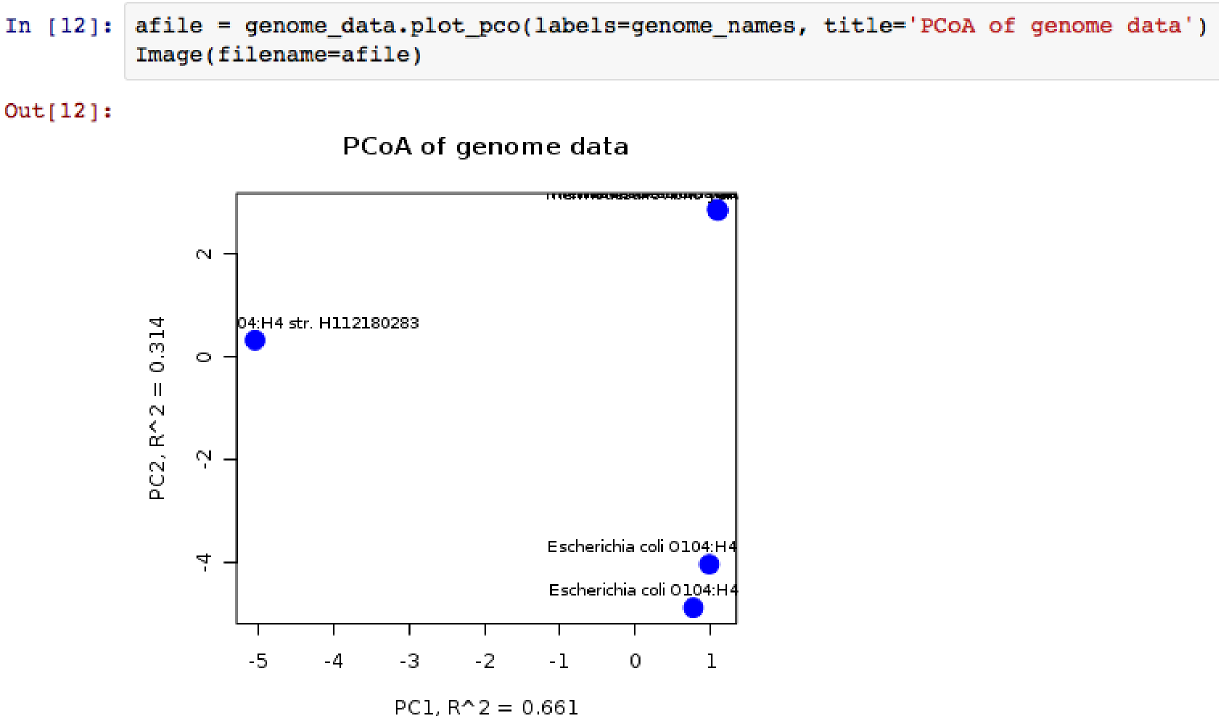
1. kbase api's / aux_store for data retrieval
2. abundance data
3. matR R package
4. normalization
5. distance matrix
6. pcoa
7. heatmap
8. plots

examples:

1. genome subsystem abundance
    a. pcoa

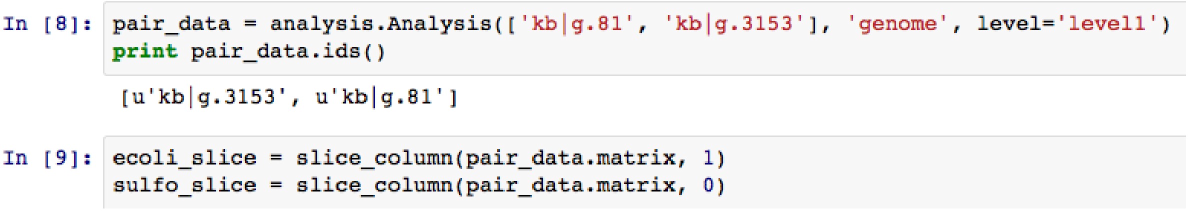


```
In [2]: genome_data = analysis.Analysis(genome_ids, 'genome', level='subsystem')
        print genome_data.ids()

        [u'kb|g.3153', u'kb|g.3387', u'kb|g.676', u'kb|g.75', u'kb|g.80', u'kb|g.81']

In [3]: genome_data.annotations()[:10]

Out[3]: [u'16S rRNA modification within P site of ribosome',
         u'2-oxoisovalerate to 2-isopropyl-3-oxosuccinate module',
         u'2-phosphoglycolate salvage',
         u'271-Bsub',
         u'5-FCL-like Experimental',
         u'5-FCL-like protein',
         u'A Gammaproteobacteria Cluster Relating to Translation',
         u'A Gram-positive cluster that relates ribosomal protein L28P to a set of uncharacte
         u'A Hypothetical Protein Related to Proline Metabolism',
         u'A Hypothetical that Clusters with PEP Synthase']
```

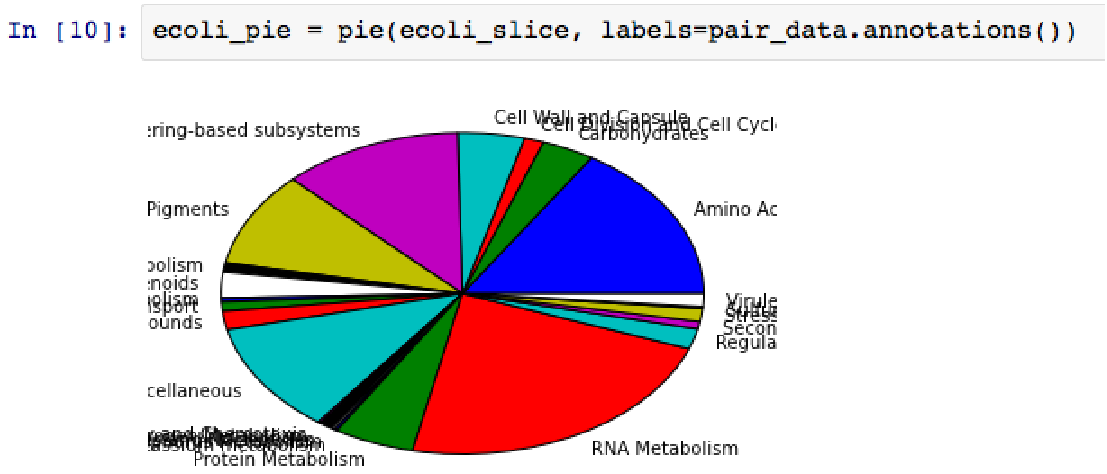


```
In [12]: afile = genome_data.plot_pco(labels=genome_names, title='PCoA of genome data')
         Image(filename=afile)

Out[12]:
```

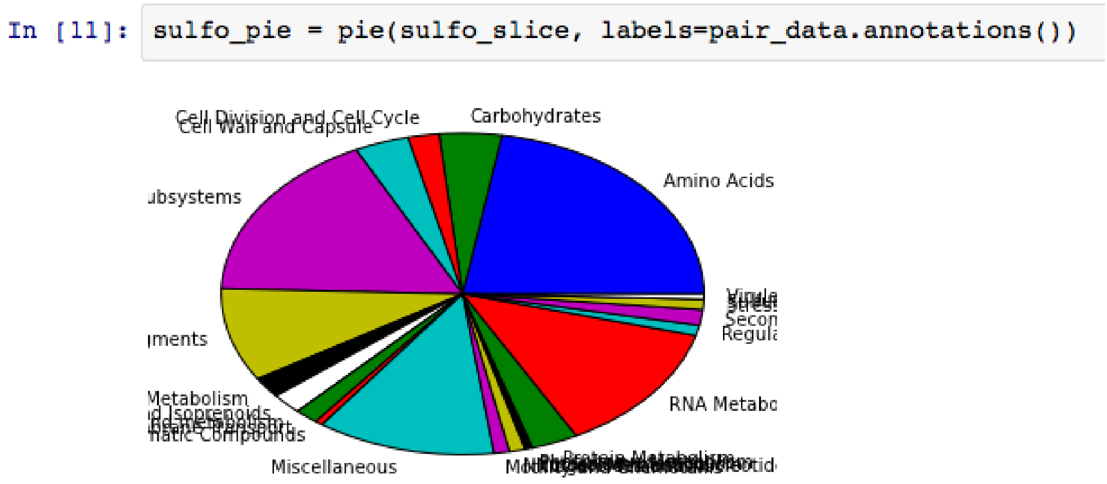
b. pie chart



```
In [8]: pair_data = analysis.Analysis(['kb|g.81', 'kb|g.3153'], 'genome', level='level1')
        print pair_data.ids()

        [u'kb|g.3153', u'kb|g.81']

In [9]: ecoli_slice = slice_column(pair_data.matrix, 1)
        sulfo_slice = slice_column(pair_data.matrix, 0)
```

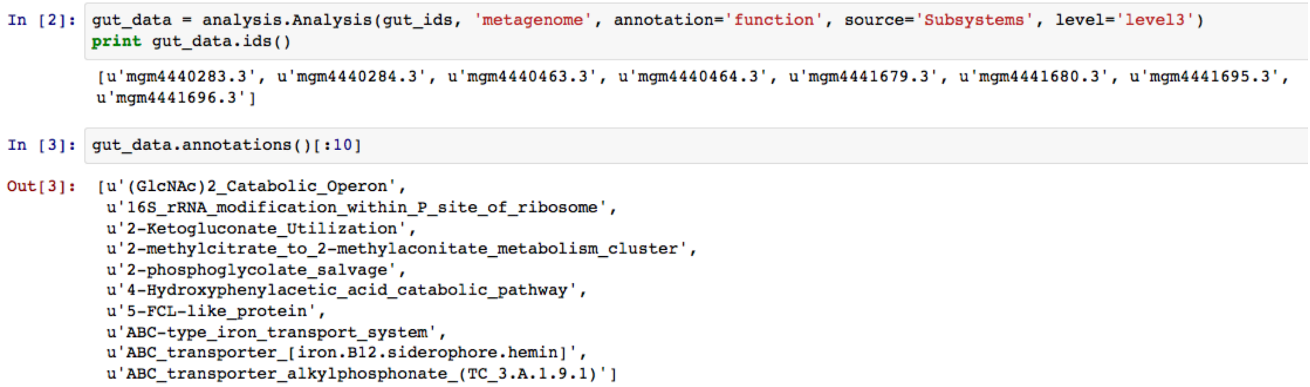


```
In [10]: ecoli_pie = pie(ecoli_slice, labels=pair_data.annotations())
```



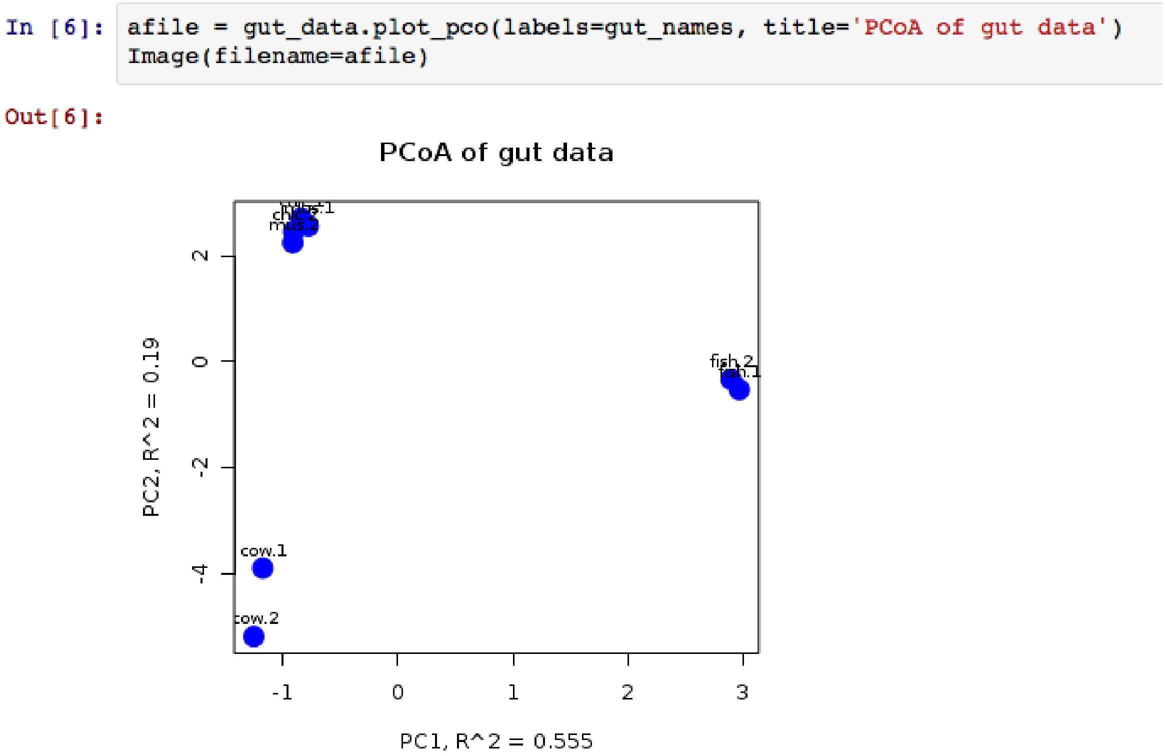
```
In [11]: sulfo_pie = pie(sulfo_slice, labels=pair_data.annotations())
```

2. community subsystem abundance



```
In [2]: gut_data = analysis.Analysis(gut_ids, 'metagenome', annotation='function', source='Subsystems', level='level3')
        print gut_data.ids()

        [u'mgm4440283.3', u'mgm4440284.3', u'mgm4440463.3', u'mgm4440464.3', u'mgm4441679.3', u'mgm4441680.3', u'mgm4441695.3',
        u'mgm4441696.3']

In [3]: gut_data.annotations()[:10]

Out[3]: [u'(GlcNAc)2_Catabolic_Operon',
         u'16S_rRNA_modification_within_P_site_of_ribosome',
         u'2-Ketogluconate_Utilization',
         u'2-methylcitrate_to_2-methylaconitate_metabolism_cluster',
         u'2-phosphoglycolate_salvage',
         u'4-Hydroxyphenylacetic_acid_catabolic_pathway',
         u'5-FCL-like_protein',
         u'ABC-type_iron_transport_system',
         u'ABC_transporter_[iron.B12.siderophore.hemin]',
         u'ABC_transporter_alkylphosphonate_(TC_3.A.1.9.1)']
```

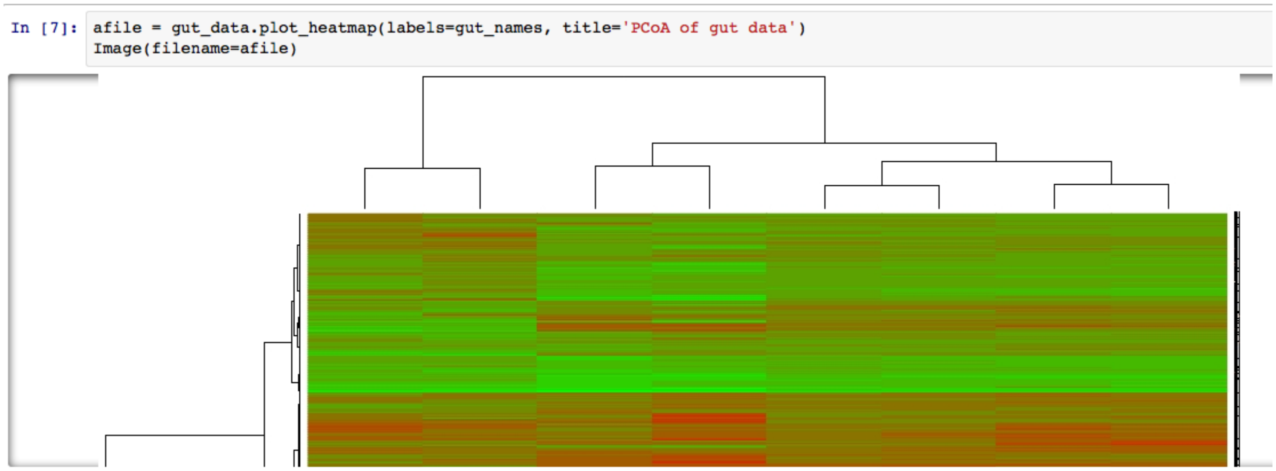
   a. pcoa



```
In [6]: afile = gut_data.plot_pco(labels=gut_names, title='PCoA of gut data')
        Image(filename=afile)
```

Out[6]:

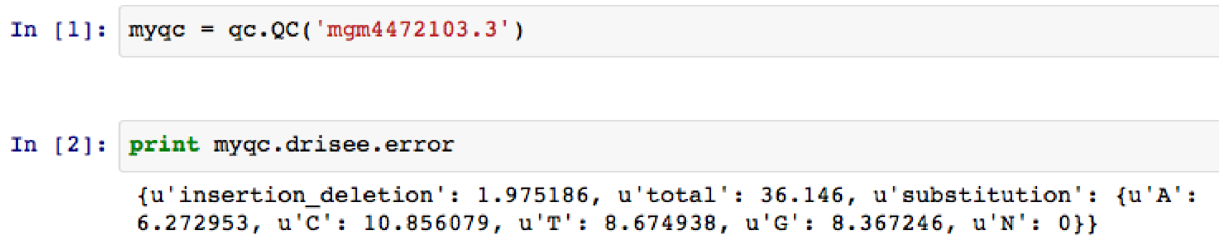
b. heatmap



```
In [7]: afile = gut_data.plot_heatmap(labels=gut_names, title='PCoA of gut data')
        Image(filename=afile)
```

QC Usage

1. create QC object from ID
   a. myqc = qc.QC(<ID>)

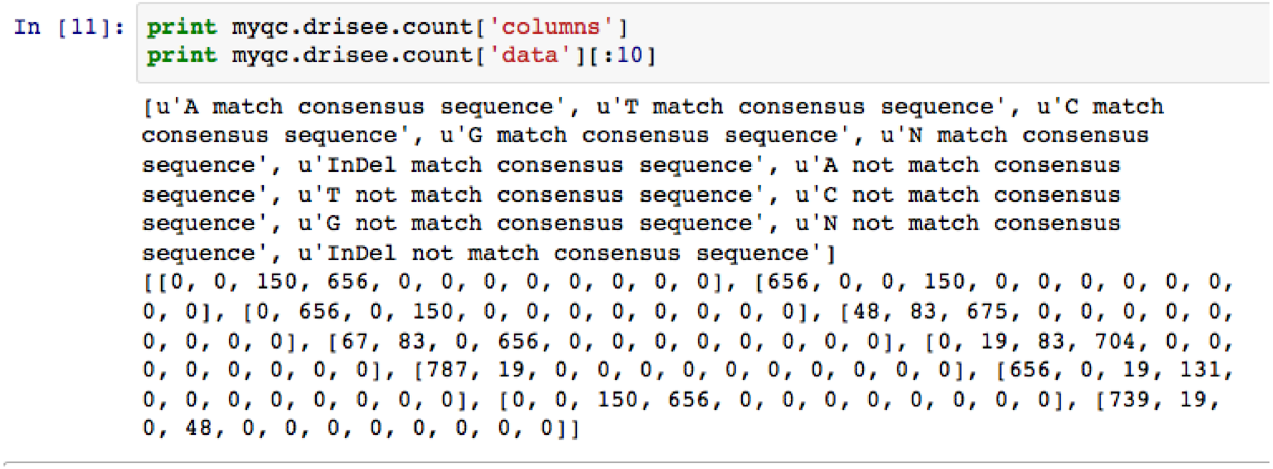


```
In [1]: myqc = qc.QC('mgm4472103.3')
```

```
In [2]: print myqc.drisee.error
```

```
{u'insertion_deletion': 1.975186, u'total': 36.146, u'substitution': {u'A':
6.272953, u'C': 10.856079, u'T': 8.674938, u'G': 8.367246, u'N': 0}}
```

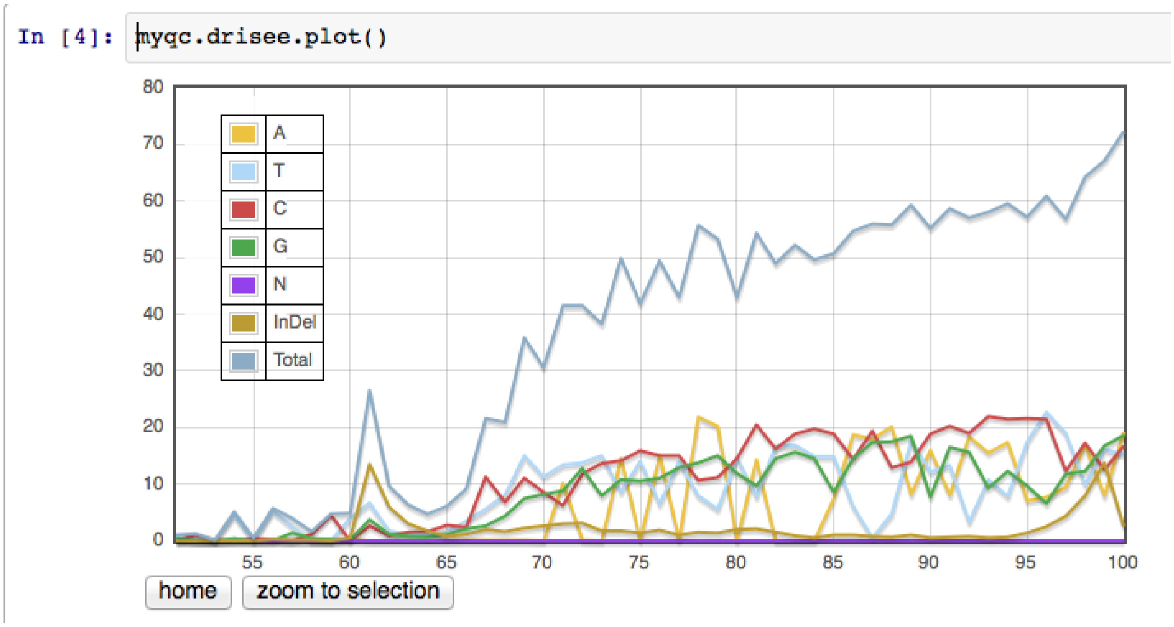
2. QC object contains 3 object types
    a. drisee
    b. kmer
    c. bp histogram
3. view data types



```
In [11]: print myqc.drisee.count['columns']
         print myqc.drisee.count['data'][:10]
```

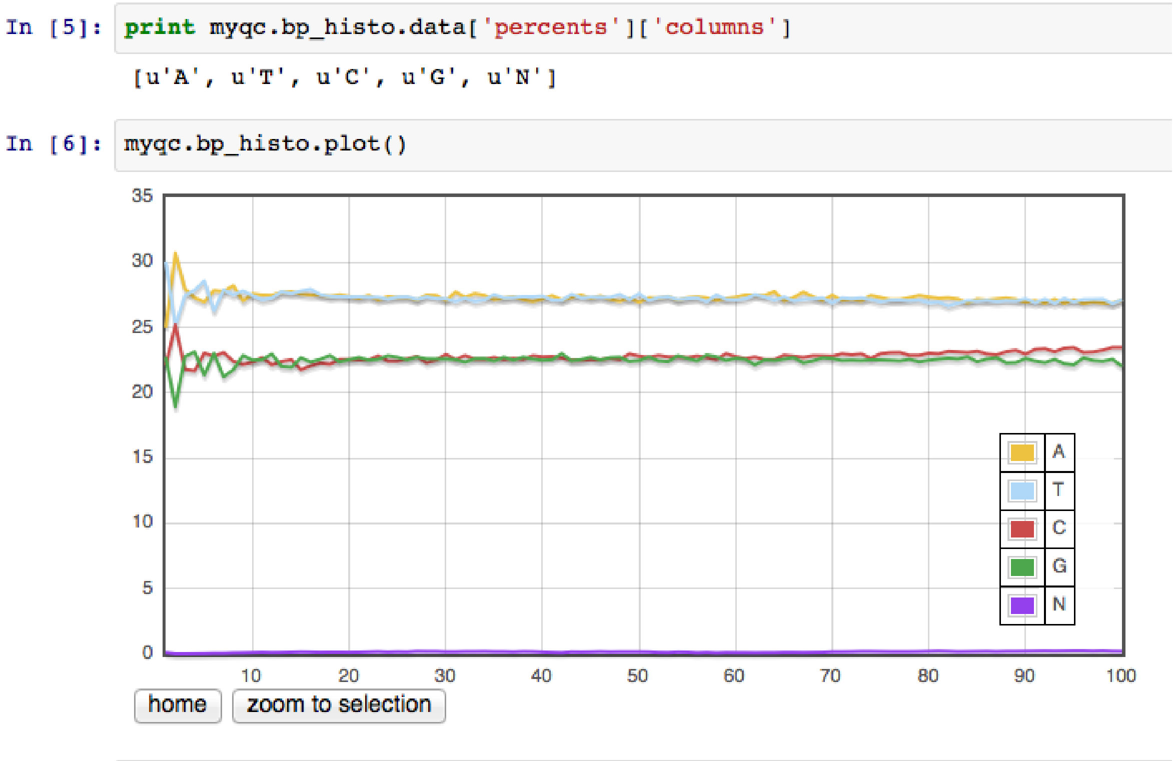
```
[u'A match consensus sequence', u'T match consensus sequence', u'C match
consensus sequence', u'G match consensus sequence', u'N match consensus
sequence', u'InDel match consensus sequence', u'A not match consensus
sequence', u'T not match consensus sequence', u'C not match consensus
sequence', u'G not match consensus sequence', u'N not match consensus
sequence', u'InDel not match consensus sequence']
[[0, 0, 150, 656, 0, 0, 0, 0, 0, 0, 0, 0], [656, 0, 0, 150, 0, 0, 0, 0, 0, 0,
0, 0], [0, 656, 0, 150, 0, 0, 0, 0, 0, 0, 0, 0], [48, 83, 675, 0, 0, 0, 0, 0,
0, 0, 0, 0], [67, 83, 0, 656, 0, 0, 0, 0, 0, 0, 0, 0], [0, 19, 83, 704, 0, 0,
0, 0, 0, 0, 0, 0], [787, 19, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [656, 0, 19, 131,
0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 150, 656, 0, 0, 0, 0, 0, 0, 0, 0], [739, 19,
0, 48, 0, 0, 0, 0, 0, 0, 0, 0]]
```

    a. print myqc.drisee.error
    b. print myqc.drisee.count
    c. print myqc.drisee.percent
    d. print mycq.kmer.data
    e. print myqc.bp_histo.count
    f. print myqc.bp_histo.percent
4. plot data types
    a. myqc.drisee.plot()



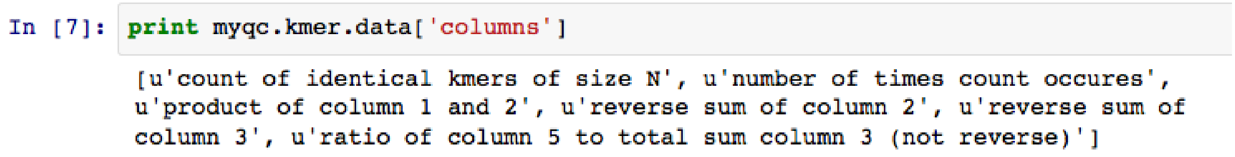
```
In [4]: myqc.drisee.plot()
```

b. myqc.bp_histo.plot()

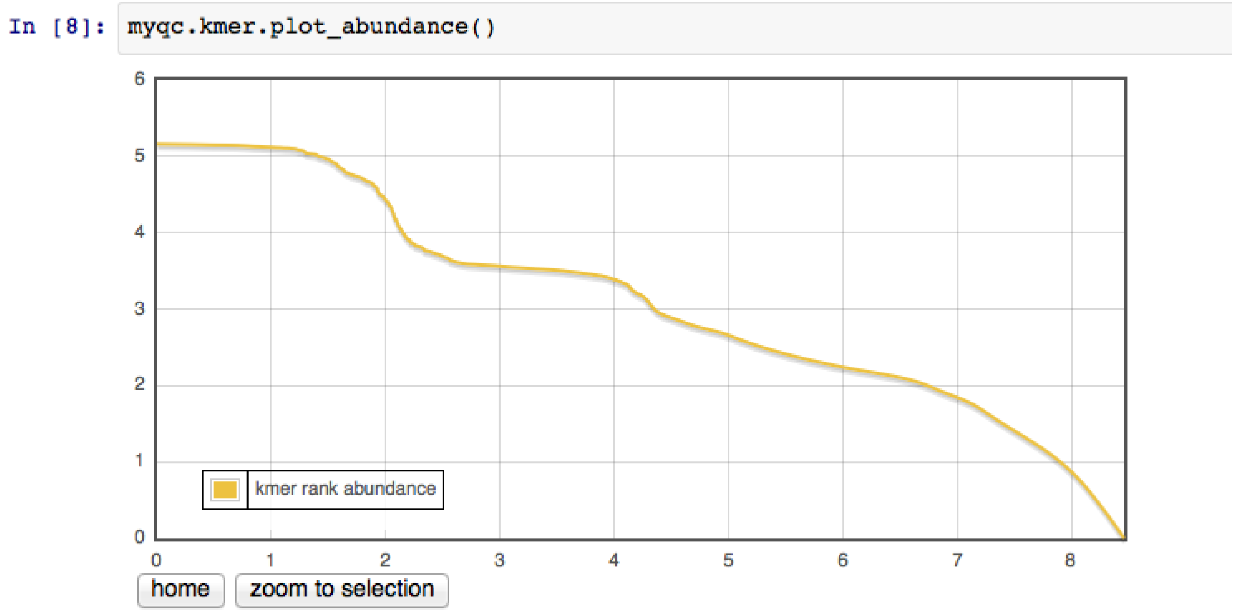


```
In [5]: print myqc.bp_histo.data['percents']['columns']

        [u'A', u'T', u'C', u'G', u'N']

In [6]: myqc.bp_histo.plot()
```

c. myqc.kmer.plot_abundance()

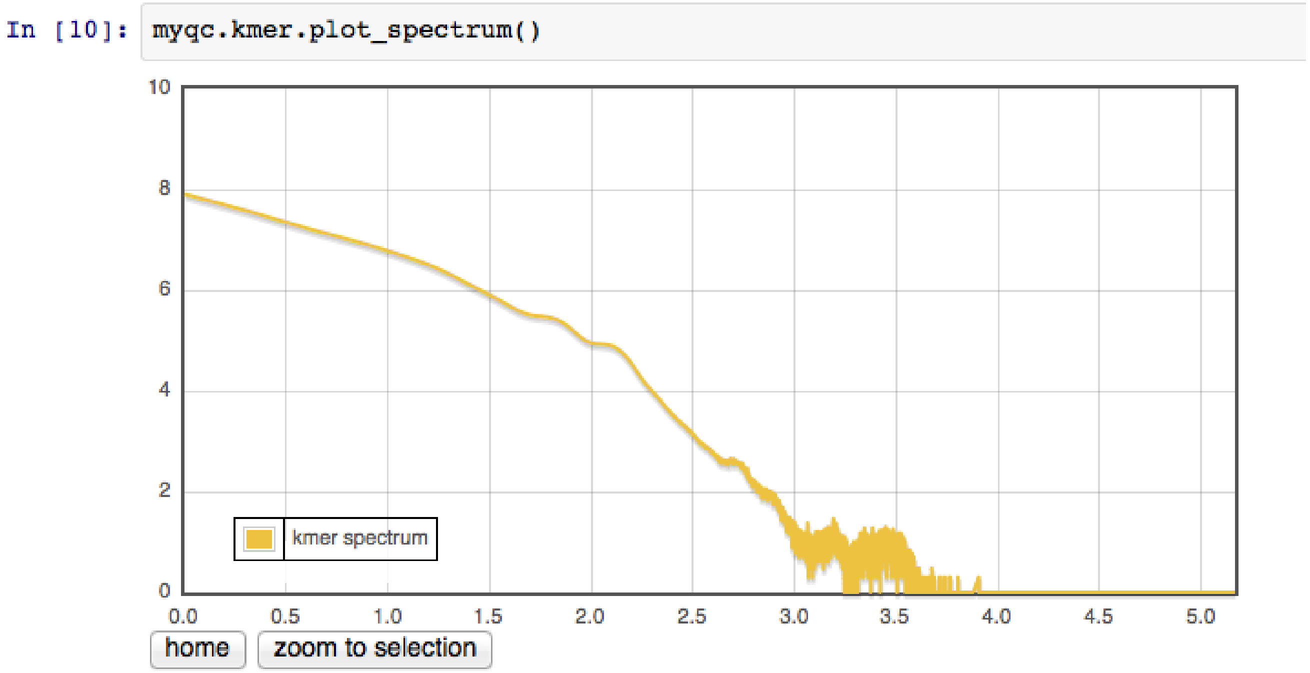


```
In [7]: print myqc.kmer.data['columns']

        [u'count of identical kmers of size N', u'number of times count occures',
        u'product of column 1 and 2', u'reverse sum of column 2', u'reverse sum of
        column 3', u'ratio of column 5 to total sum column 3 (not reverse)']
```



```
In [8]: myqc.kmer.plot_abundance()
```

d. myqc.kmer.plot_ranked()
e. myqc.kmer.plot_spectrum()



```
In [10]: myqc.kmer.plot_spectrum()
```