

This pseudocode is showing how we want to use pyjugex in a python notebook.
Version 1 wouldnt require HBP atlas API.

```
import pyjugex
from nibabel import Nifti1Image
import numpy as np

refresh_cash=False
gene_cash="/home/myself/.pyjugex/cache"

# specify an atlas, and select two areas
parcels = Nifti1Image.open('mpm.nii.gz','r').get_data()
atlasregions = np.loadtxt('mpm_labels.txt')
l0 = atlasregions['V1'] # this would give as an index, e.g. 130
l1 = atlasregions['V2'] # this would give as an index, e.g. 132

# specify candidate genes and trigger the workflow
analysis = pyjugex.DifferentialGeneExpression()
entrez_ids = [6494,9123,1892]
if refresh_cache=True:
    # this takes a while
    analysis.retrieve_gene_data(entrez_ids,cache=gene_cash)
analysis.set_candidate_genes(entrez_ids,cache=gene_cash)
analysis.set_coordinates_region1(np.where(parcels==l1))
analysis.set_coordinates_region2(np.where(parcels==l2))
# this takes several minutes
analysis.run()

# see the results
result = analysis.pvalues()
print result[6494] # gives us the pvalue
print [id for id in entrez_ids if result[id]>.05]
```

Version 2 would use a pythonic atlas API.

```
from hbp.atlas.human import jubrain
xyz_l0 = jubrain.coordinates(jubrain.region_names['V1'])
xyz_l1 = jubrain.coordinates(jubrain.region_names['V2'])
..
analysis.set_coordinates_region1(xyz_l0)
```