

A complete Python application in 143 lines

Iain Bancarz
ib5@sanger.ac.uk
23 September 2014

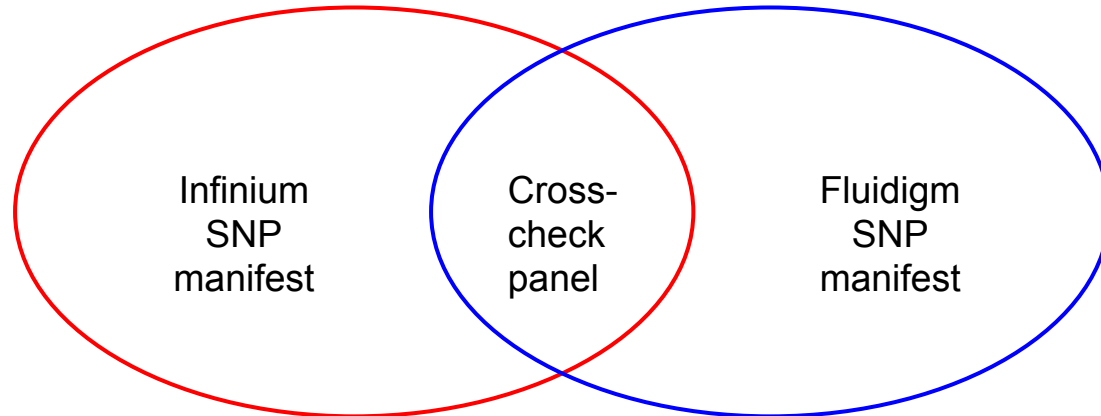
What do I mean by “complete”?

- Does something useful
- Object-oriented
- Includes:
 - Tests
 - Command-line interface

```
ib5@farm3-head3:scripts$ wc -l *.py
100 rename_infinium_as_fluidigm.py
 43 test.py
143 total
ib5@farm3-head3:scripts$ █
```

The task

- SNP names in manifests must be consistent
- Will be used to cross-check results (genotype calls) on different platforms



Updating the Infinium manifest

Before:

```
3,exm-IND1-85310248,1,85537661,0.8554,[I/D],PLUS,PLUS,1
4,exm-IND10-102817747,10,102827758,0.7945,[I/D],PLUS,PLUS,2
5,exm689414,8,23060256,0.9067,[T/C],BOT,BOT,2
6,exm854649,10,106039185,0.7872,[A/G],TOP,TOP,2
```

After:

```
3,exm-IND1-85310248,1,85537661,0.8554,[I/D],PLUS,PLUS,1
4,exm-IND10-102817747,10,102827758,0.7945,[I/D],PLUS,PLUS,2
5,rs6557634,8,23060256,0.9067,[T/C],BOT,BOT,2
6,rs156697,10,106039185,0.7872,[A/G],TOP,TOP,2
```

Executable script with classes

```
class SNPNameConverter:

    def __init__(self, verbose):
        self.verbose = verbose

    def convert(self, infinumPath, fluidigmPath, outPath):
        snps = self.readFluidigm(fluidigmPath)
        if self.verbose:
            msg = "Read %d Fluidigm SNPs\n" % (len(snps.keys()),)
            sys.stderr.write(msg)
        (modified, total) = self.readWriteInfinum(infinumPath, outPath, snps)
        if self.verbose:
            msg = "Renamed %d of %d Infinum SNPs\n" % (modified, total)
            sys.stderr.write(msg)
```

```
if __name__ == "__main__":
    main()
```

The Python argparse module

- Fully functional command-line interface in 8 lines!

```
desc = "Script to rename SNPs in an Infinium manifest for consistency with the  
equivalent SNPs in Fluidigm."  
parser = ArgumentParser(description=desc)  
parser.add_argument('-i', '--infinium', help='Path to an Infinium .bpm.csv SNP  
manifest file. Required.', required=True, metavar="PATH")  
parser.add_argument('-f', '--fluidigm', help='Path to a Fluidigm .tsv SNP man  
ifest file. Required.', required=True, metavar="PATH")  
parser.add_argument('-o', '--out', help='Path for an output Infinium .bpm.csv  
file. Required unless --overwrite is in effect.', metavar="PATH")  
parser.add_argument('--overwrite', help='Overwrite the input .bpm.csv file wi  
th the output. Incompatible with --out.', action='store_true')  
parser.add_argument('-v', '--verbose', help='Print status information to STDE  
RR', action='store_true')  
args = vars(parser.parse_args())
```

```
ib5@farm3-head3:scripts$ python rename_infinium_as_fluidigm.py --help
usage: rename_infinium_as_fluidigm.py [-h] -i PATH -f PATH [-o PATH]
                                     [--overwrite] [-v]
```

Script to rename SNPs in an Infinium manifest for consistency with the equivalent SNPs in Fluidigm.

optional arguments:

| | |
|--------------------------|--|
| -h, --help | show this help message and exit |
| -i PATH, --infinium PATH | Path to an Infinium .bpm.csv SNP manifest file. Required. |
| -f PATH, --fluidigm PATH | Path to a Fluidigm .tsv SNP manifest file. Required. |
| -o PATH, --out PATH | Path for an output Infinium .bpm.csv file. Required unless --overwrite is in effect. |
| --overwrite | Overwrite the input .bpm.csv file with the output. Incompatible with --out. |
| -v, --verbose | Print status information to STDERR |

```
ib5@farm3-head3:scripts$ █
```

Testing with Python unittest

```
import os, unittest
from shutil import rmtree
from tempfile import mkdtemp
from rename_infinium_as_fluidigm import SNPNameConverter

class TestConverter(unittest.TestCase):

    def setUp(self):
        dataDir = './data'
        self.infinium = dataDir+'/test_infinium_input.bpm.csv'
        self.fluidigm = dataDir+'/qc_fluidigm_snp_info_GRCh37.tsv'
        self.refOutput = dataDir+'/test_infinium_output.bpm.csv'
        self.tempdir = mkdtemp()
        self.outPath = self.tempdir+'/output.bpm.csv'

    def tearDown(self):
        rmtree(self.tempdir)

    def test_class(self):
        verbose = False
        converter = SNPNameConverter(verbose)
        converter.convert(self.infinium, self.fluidigm, self.outPath)
        outData = open(self.outPath).read()
        refData = open(self.refOutput).read()
        self.assertEqual(outData, refData)
```


The application

- Available in Sanger internal git repository:
`http://git.internal.sanger.ac.uk`
- **Project** `gapi/genotyping manifests`
- `http://git.internal.sanger.ac.uk/cgi-bin/gitweb.cgi?p=gapi/genotyping_manifests.git;a=summary`