Power Analyses for SEQLinkage

Wed Apr 2 10:45:41 CDT 2014 | Let's rock and roll!

Basics

We simulate and analyze two autosomal recessive genes GJB2 and SLC26A4 and two autosomal dominant genes MYO7A and MYH9. The program takes two genes source data as input.

As we call `seqlink` from inside of the program multiple times we'd like to speed up the execution by using ramdisk of 8000MB.

```
sudo mkdir -p /ramcache
sudo mount -t tmpfs ramfs -o size=8000m /ramcache
cd /ramcache
ln -s ~/SVN/SEQLinco/simulations/LinkagePowerCalc.py pcal.py
```

Hierarchy of simulation parameters

I want to draw one figure with 4 panels. Each panel is one mode of inheritance, namely A&B are recessive and compound recessive using recessive genes; C&D are dominant and compound dominant using recessive genes. On each panel, the X-axis is sample size and Y-axis is allelic heterogeneity. Powers are contour plots of two colors, one color for new method, one for old method. The only parameter not scanned here is `--offsprings`. For now I set it in between 3 to 8. It can be varied for additional simulations in supplemental. In this figure only one of the gene pair will be reported and the complementary figure can be found in supplemental.

Linkage analysis parameters are always `-K 0.01 -W 0 -M 1`. See `seqlink -h` for details.

Example command

```
python pcal.py -g MYH9.tsv MYO7A.tsv -m dominant -r 10 --sample-size 20 \
--blueprint blueprint.txt --vanilla --tempdir /ramcache \
--run-linkage -K 0.01 --moi AD -W 0 -M 1 --output-entries 0
```

Power calculation script

Script generator:

PYTHON

```
name = 'pcal.sh'
rep = 100
batch = 50
count = 0
gdict = {'recessive': 'SLC26A4.tsv GJB2.tsv', 'compound_recessive':'SLC26A4.tsv GJB2.tsv', 'dominant':'MYO7A.tsv MYH9.tsv', 'compoun
d_dominant':'MYO7A.tsv MYH9.tsv'}
mdict = {'recessive':'AR', 'compound_recessive':'AR', 'dominant':'AD', 'compound_dominant':'AD'}
fdict = {' ': 'CHPResult.csv', '--single-markers': 'SNVResult.csv'}
with open(name, 'w') as f:
```

```
    for m in ['recessive', 'dominant', 'compound_recessive', 'compound_dominant']:
        for a in range(0, 110, 10):
            a = a * 0.01
            for s in range(5, 60, 5):
                for method in [' ', '--single-markers']:
                    count += 1
                    f.write("./pcal.py --debug -n 3 8 -g {0} -m {1} -s {2} -a {3} {4} -r {5} --ofile {9} " \
                          "--blueprint blueprint.txt --vanilla {6} --jobs 1 --run-linkage -K 0.01 " \
                          "-W 0 -M 1 --moi {7} --output-entries 0 --tempdir /ramcache/tmp >> {8}.{9}\n".\
                          format(gdict[m], m, s, a, 1-a, rep, method, mdict[m], fdict[method], "RUN%s" % count))
                    if count % batch == 0:
                       f.write("echo {}\n".format(count))
```

Use command below to run the generated script:

```
cat pcal.sh | gw-parallel -j7
```

Power calculation result database

```
cat CHPResult.csv.* > CHPResult.csv
cat SNVResult.csv.* > SNVResult.csv
./pcal.py --print-header | wsqlite PowerCalc -i CHPResult.csv --as CHP --header - -d","
./pcal.py --print-header | wsqlite PowerCalc -i SNVResult.csv --as SNV --header - -d","
```

Now the power calculation results are in the database

```
wsqlite PowerCalc -s
wsqlite PowerCalc -s CHP
wsqlite PowerCalc -s SNV
```

Let's have a rough feeling on power comparison between the methods

BASH

```
for i in recessive dominant compound_recessive compound_dominant; do
    echo $i
    wsqlite PowerCalc "select avg(plod1) from CHP where moi = '$i'"
    wsqlite PowerCalc "select avg(plod1) from SNV where moi = '$i'"
done
```

OUTPUT

```
recessive
0.727776074443
0.746848342238
dominant
0.528512396694
0.656528925619
compound_recessive
0.731162035468
0.438850488354
compound_dominant
0.527520661156
0.521157024793
```