

BonnGWAS all pairs!

Organizando SNPs de T. Becker

1.- Primero se ponen en una linea las parejas

```
for x in list*.txt; do cat $x | awk {'print $1'} >> ${x%.txt}.cont; cat $x |
awk {'print $2'} >> ${x%.txt}.cont; sort ${x%.txt}.cont > ${x%.txt}.sc; rm
${x%.txt}.cont; done
```

😬 Que estupidez!!!!!! Asi mejor:

```
$ for x in list*.txt; do cat $x | awk {'print $1"\n"$2'} | sort | uniq >
${x%.txt}.sc; done
```

2.- Con este nuevo archivo se buscan todos los proxies posibles de cada marcador en <http://www.broadinstitute.org/mpg/snap/ldsearch.php> con estos datos,

```
SNP data set: 1000 Genomes Pilot 1
r2 threshold: 0.7
Population panel: CEU
Distance limit: 100
```

se salva el resultado con la extension .proxies

3.- Ahora se escriben las parejas de proxies posibles utilizando la lista de parejas original (x.txt) y la lista de proxies de cada marcador (x.proxies)

```
$ find_pairs.pl x.txt
```

construye las parejas de proxies tomando las parejas originales de x.txt y los proxies de cada marcador de x.proxies. Escribe el resultado en x.pairs.

[find_pairs.pl](#)

```
use strict; use warnings;
use File::Slurp qw(read_file);
use List::MoreUtils qw(uniq);

sub get_proxies{
    (my $rs, my $pfile) = @_;
    my $patt = '^' . $rs . '\s+(rs\d{1,18})\s+(\d{1,6})\s+(\d{1}\.\d{3})';
    my @proxies = map { /$patt/; $1 } grep {/^$rs/} read_file $pfile;
    return @proxies;
}

my $ifile = shift;
die "Must supply input filename!\n" unless ($ifile);
(my $bn) = $ifile =~ /^(.*)\.\d{3}$/;
my $pfile = $bn . '.proxies';
```

```

my $ofile = $bn.'.pairs';
my $tl = int qx/wc -l $ifile | awk {'print $1'};/;
open(IPF, "<$ifile") or die "Cant open input file\n";
my @pairs;
print "looking for pairs in $tl inputs...\n";
my $l = 0;
while(<IPF>){
    (my $a, my $b) = /(rs\d{1,18})\s+(rs\d{1,18})/;
    $l++;
    my $prc = 100*$l/$tl;
    $prc =~ s/^\(\d{1,3}\.\d{2}\)\d*$/$1/;
    print "$a - $b -> $prc%\n";
    if($a && $b){
        my @aps = uniq get_proxies($a, $pfile);
        my @bps = uniq get_proxies($b, $pfile);
        foreach my $ap (sort @aps){
            foreach my $bp (sort @bps){
                push @pairs, "$ap $bp";
            }
        }
    }
}
close IPF;

my @upairs = uniq @pairs;
my $np = scalar @upairs;

print "writing $np pairs to $ofile ...\n";
open(OPF, ">$ofile") or die "Cant open output file\n";
foreach my $pair (sort @upairs){
    print OPF "$pair\n";
}
close OPF;

```

4.- Despues se escogen sólo las parejas donde ambos marcadores esten descritos en la base de datos. Ejemplo,

```

grppairs.pl listTEST5.pairs /home/data/Variomics/ADNIimpQC2.bim >
listTEST5.pairs.grp

```

grppairs.pl

```

use strict; use warnings;
use File::Slurp qw(read_file);

my $sfile = shift;
my $dbfile = shift;

my %series = map {/^(rs\d{1,18})\s+(rs\d{1,18})$/, $1=>$2} read_file

```

```

$file;
my %dbpairs = map {/\s+(rs\d{1,18})\s+/, $1=>1} grep {/rs\d{1,18}/}
read_file $dbfile;

foreach my $pair (sort keys %series){
    if(exists $dbpairs{$pair} && exists $dbpairs{$series{$pair}}){
        print "$pair $series{$pair}\n";
    }
}

```

5.- Tras esto ya se puede correr intersnp de una manera racional.

Para los dos últimos pasos hemos hecho un script que engloba el procesamiento de un test en todas las DBs.

[isnp4all.pl](#)

```

use strict; use warnings;
use File::Basename qw(basename);
use File::Find::Rule;

my $origpairs = shift;
my @dbs = qw(ADMURimpQC2 ADNIimpQC2 GenADA_impQC2 NIA_AD_impQC2
TGEN_impQC2);
my $sel_template = '/home/data/Bonn_GWAS/selection_file_template.txt';
(my $test) = $origpairs =~ /listTEST(\d*)\.pairs/;
my $w_dir = 'intersnp_TEST' . $test;
mkdir $w_dir;
chdir $w_dir;
my $odir = 'outputs';
mkdir $odir;
my $tmpsel = 'selection_file_TEST' . $test . '.txt';
foreach my $dbname (@dbs){
    my $targetdb = '/home/data/Variomics/' . $dbname;
    my $grpfile = 'listTEST' . $test . '_' . $dbname . '.grp';
    my $order = 'grppairs.pl ' . $origpairs . ' ' . $targetdb . '.bim >
' . $grpfile;
    print "Grep $origpairs into $dbname...\n";
    system($order);
    print "Done\n";
    open TEMPLATE, "<$sel_template" or die "Could not find template\n";
    open TMPS, ">$tmpsel";
    my $ofile = $odir . '/' . $dbname . '_TEST' . $test . '_';
    while (<TEMPLATE>){
        s/<Database>/$targetdb/;
        s/<Test>/$test/;
        s/<Pairs>/$grpfile/;
        s/<Output>/$ofile/;
        print TMPS "$_";
    }
}

```

```
close TMPS;  
close TEMPLATE;  
# my $order = 'intersnpPN '.$tmpsel;  
# system($order);  
$order = 'intersnp '.$tmpsel;  
print "$order\n";  
system($order);  
}
```

Para ello utilizamos la plantilla,

```
$ cat selection_file_template.txt  
BFILE <Database> // put path and name of plink bfile there  
TWO_MARKER 1  
PRINTTOP 50000  
TEST <Test>  
COMBIFILE <Pairs> // if file is in working dir, else put path  
COMBILIST 1  
OUTPUTNAME <Output>  
END
```

From:

<https://imagen.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:

https://imagen.fundacioace.com/wiki/doku.php?id=genetica:bonngwas_all

Last update: **2020/08/04 10:58**

