AgileROHFinder and AgileROHFilterer

Command line applications for the detection of autozygous regions from Exome or Affymetrix microarray SNP data.

Data format

AgileROHFinder identifies autozygous regions using genotype data formatted as either VCF files or Affymetrix genotype files (or older xls or birdseed) text files. Similarly, AgileROHFilterer will process a VCF file (but not microarray genotype data), creating a second VCF that only contains variants within the autozygous regions. The format and required fields of the VCF and microarray input files is described here.

Note:

> These programs will process a specified sample in a VCF file containing multiple samples, but will only process the last sample in Affymetrix genotype data files.

Creating the programs

The source code can be compiled on both Linux and Windows computers as described here.

Precompiled programs

Both programs have been precompiled for Linux and Windows and placed in the Program folder. Select the appropriate OS and download the programs (Linux or Windows). Due to the security policies of some organisations, downloading programs on a Windows computer may not be straight forward, however this guide may help. Due to variations in the set up of Linux systems the Linux programs may not work and so may need to be compiled from the source code.

Running the programs

These programs are console applications and so do not have a user interface. They run within a terminal environment. On Linux this will typically be in a bash terminal while on windows it will be a "Command Prompt" or "PowerShell" terminal. If the analysis is preformed on a remote server the application would typically be run from the inbuilt bash terminal on Linux or Mac or on windows via a third party terminal such as Putty. In both cases they connect to the remote server via an SSH connection.

The examples below use the Linux file structure were /data/in.vcf refers to a file in the "data" folder, the equivalent on Windows would be "C:\data\in.vcf".

Helpful scripts

The Program > scripts page contains a python and bash scripts that may be helpful.

Commands

Both programs require similar commands, the structure of the commands (without optional parameters) to run `AgileROHFinder` and `AgileROHFilterer` are shown below and described in greater detail in the table.

Basic commands with on optional parameters

`AgileROHFinder`

> /path/AgileROHFinder.exe /data/in.vcf /data/out.txt -t

`AgileROHFilterer`

> /path/AgileROHFilterer.exe /data/in.vcf /data/out.vcf /data/out.txt 500000 -t

Command with all optional parameters

`AgileROHFinder`

> /path/AgileROHFinder.exe /data/in.vcf /data/out.txt -t -V -Y 2

`AgileROHFilterer`

> /path/AgileROHFilterer.exe /data/in.vcf /data/out.vcf /data/out.txt 500000 -t -V -Y 2

-V indicates the genotypes in the VCF are to be used, ***-Y*** includes variants without a RS ID and the final ***'2'*** leads to the second sample's data to be processed in a multi-sample VCF file.

Note

> If a folder or file name contains a space the file name and its location must be placed in speech marks i.e /my data/my file.vcf should be entered as "/my data/my file.vcf"

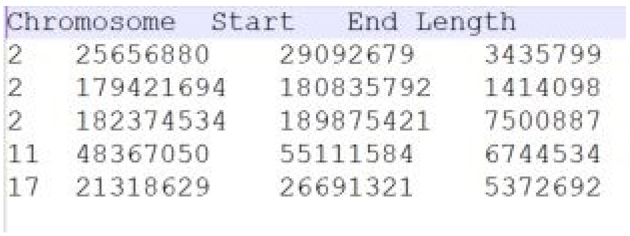
Table 1: Description of command parameters and options

Command fragment	Description	Note
/path/AgileROHFinder.exe or /path/AgileROHFilterer.exe	Name of the program with it's location	
/data/in.vcf	The name (with location) of the data file to process.	`AgileROHFilterer` will only process vcf files while `AgileROHFinder` will process vcf files and Affymetrix microarray genotype files
/data/out.vcf (`AgileROHFilterer` only)	Name of a file to save the filtered variant data too.	***Note*** This option is only present in `AgileROHFilterer` While it will create this file, it will not create any directories, so the path to the location most exist before the program is run.

Command fragment	Description	Note
/data/out.txt	The name with location of the file to save the list of autozygous regions too	While they will create this file, they will not create any directories, so the path to the location most exist before the program is run.
Any whole positive number (`AgileROHFilterer` only)	The reported regions are be extended by this number of bases when `AgileROHFilterer` filters the variants by position, such that variants just outside a region are also retained	***Note*** This option is only present in `AgileROHFilterer`
Export format options **-t**, **-b** or **-a**	Sets the format of the data results file	See Tables 2 to 4 for examples
Process all variants: **-Y**	By default only variants with an RS ID are processed, if **-Y** is set then all SNPs (with one alternative allele) will be used	**Optional**: only affects analysis of VCF data
Use genotypes in VCF file: **-V**	By default a variant's genotype is calculated by the program, if **-V** is setused, the genotype in the VCF file is used (The file most have the 'GT' field)	**Optional**: only affects analysis of VCF data
Select sample to be process in multi-sample VCF file: any whole positive number	If a VCF file has multiple samples this optional parameter sets which sample is analysed.	**Optional**: only affects analysis of VCF data

Table 2

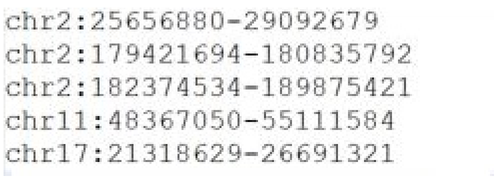
Autozygous regions output file format: Option -t (columns separated by tab character)



```
Chromosome  Start    End Length
2    25656880     29092679     3435799
2    179421694    180835792    1414098
2    182374534    189875421    7500887
11   48367050     55111584     6744534
17   21318629     26691321     5372692
```

Table 3

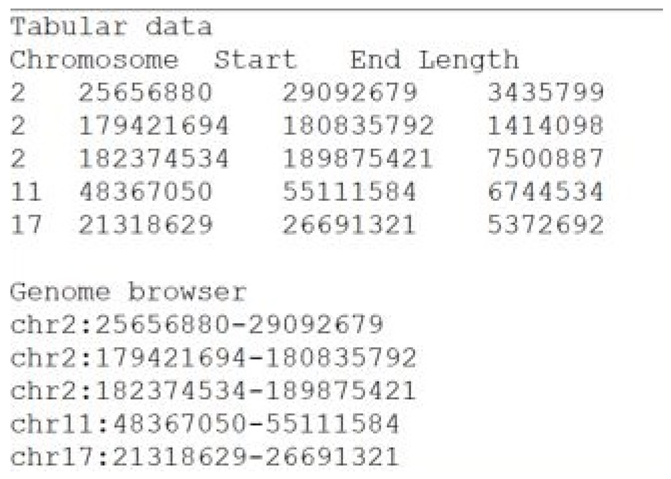
Autozygous regions output file format: Option -b (Each line can be entered in to the UCSC genome browser)



```
chr2:25656880-29092679
chr2:179421694-180835792
chr2:182374534-189875421
chr11:48367050-55111584
chr17:21318629-26691321
```

Table 4

Autozygous regions output file format: Option -a (Contains both formats)



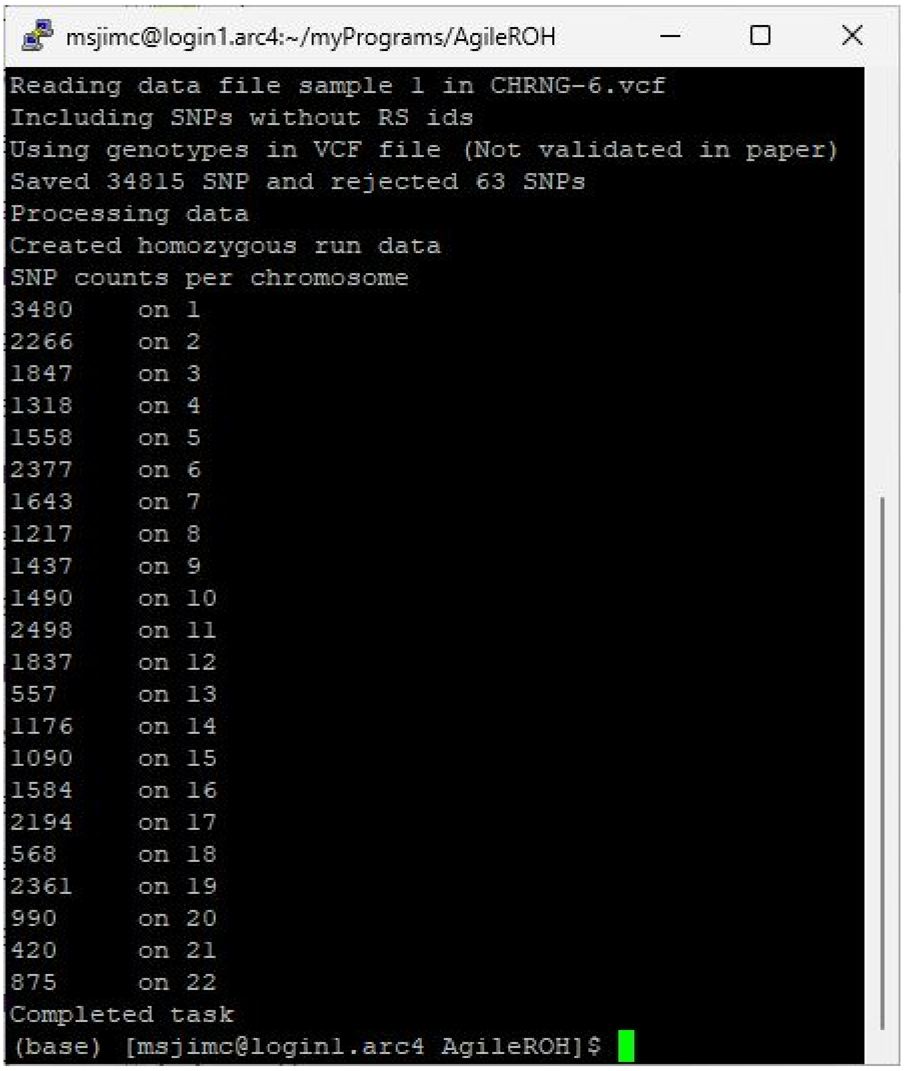
```
Tabular data
Chromosome  Start    End Length
2    25656880     29092679     3435799
2    179421694    180835792    1414098
2    182374534    189875421    7500887
11   48367050     55111584     6744534
17   21318629     26691321     5372692

Genome browser
chr2:25656880-29092679
chr2:179421694-180835792
chr2:182374534-189875421
chr11:48367050-55111584
chr17:21318629-26691321
```

Feedback

As the programs run, the current status will be shown in the terminal window.

A successful analysis:

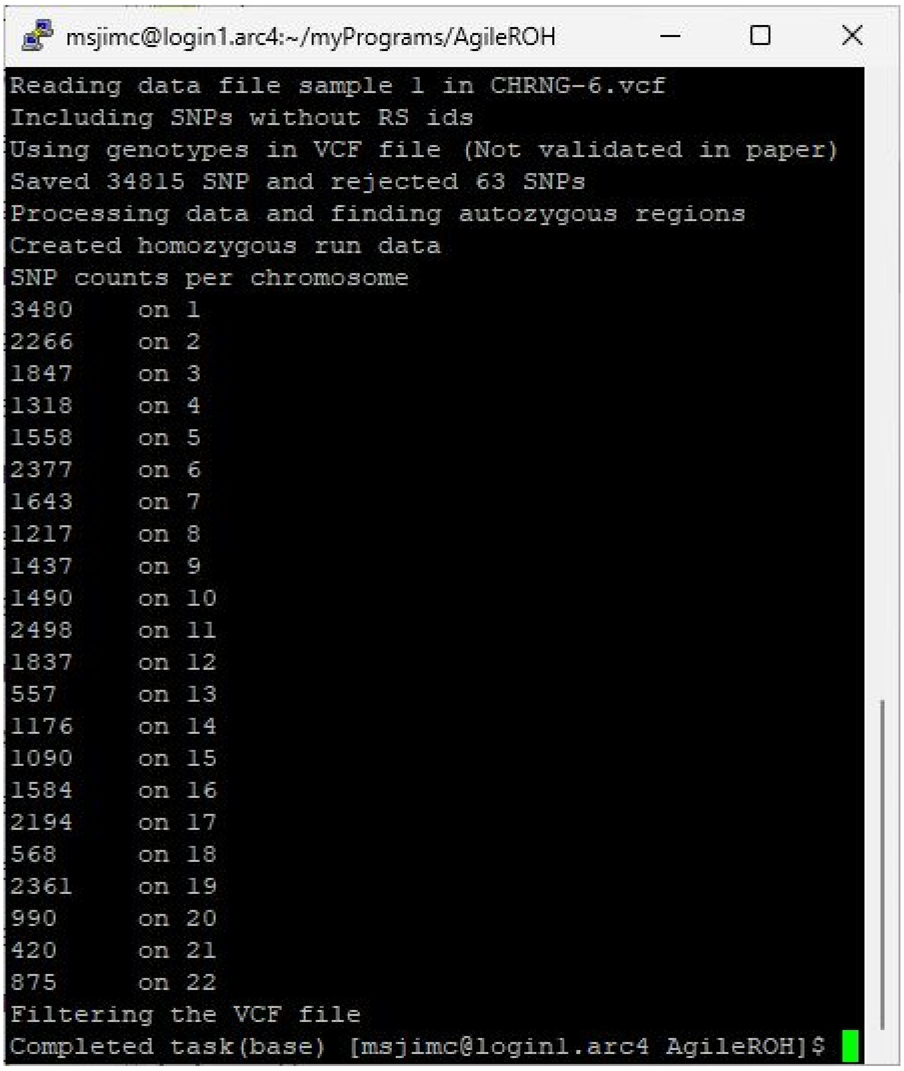
Figure 1: `AgileROHFiinder`



msjimc@login1.arc4:~/myPrograms/AgileROH

```
Reading data file sample 1 in CHRNG-6.vcf
Including SNPs without RS ids
Using genotypes in VCF file (Not validated in paper)
Saved 34815 SNP and rejected 63 SNPs
Processing data
Created homozygous run data
SNP counts per chromosome
3480    on 1
2266    on 2
1847    on 3
1318    on 4
1558    on 5
2377    on 6
1643    on 7
1217    on 8
1437    on 9
1490    on 10
2498    on 11
1837    on 12
557     on 13
1176    on 14
1090    on 15
1584    on 16
2194    on 17
568     on 18
2361    on 19
990     on 20
420     on 21
875     on 22
Completed task
(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 2: `AgileROHFilterer`



```
Reading data file sample 1 in CHRNG-6.vcf
Including SNPs without RS ids
Using genotypes in VCF file (Not validated in paper)
Saved 34815 SNP and rejected 63 SNPs
Processing data and finding autozygous regions
Created homozygous run data
SNP counts per chromosome
3480    on 1
2266    on 2
1847    on 3
1318    on 4
1558    on 5
2377    on 6
1643    on 7
1217    on 8
1437    on 9
1490    on 10
2498    on 11
1837    on 12
557     on 13
1176    on 14
1090    on 15
1584    on 16
2194    on 17
568     on 18
2361    on 19
990     on 20
420     on 21
875     on 22
Filtering the VCF file
Completed task(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 1

Figures 1 and 2 show a typical status report of the analysis of a exome vcf file by `AgileROHFinder` and `AgileROHFilterer` respectively where only variants with an RS id were used and their genotypes were calculated by the programs.

Output description:

- Initially, the program states which file is being process along with the sample's index (VCF files only).
- Next it states whether it will process variants without an RS ID. "*Including SNPs without RS ids*" indicates it will process all variants, while "*Ignoring SNPs without RS IDs*" indicates unnamed variants will be excluded.
- The program then declares whether it is calculating the genotypes from the allele read depths (Calculating genotypes) or using those in the VCF file (Using genotypes in VCF file (Not validated in paper)): Using genotypes in the VCF file was not validated in the linked paper.
- Once the file has been read, the program displays the number of SNPs saved and the number rejected. Only single base SNPs on the autosomal chromosomes are counted with the main reasons for a variant been rejected are low total read count or skewed allele read ratios.
  (Issues with the file format may also cause the SNPs to be rejected, in this case an excessive number or all the SNPs may be rejected.)

- Next the program states that it is analysing the SNP data to find autozygous regions ("*Processing data and finding autozygous regions*") followed by "*Created homozygous run data*" when the analysis is completed.
- The next 23 lines form a table of the number of SNPs analysed on each autosomal chromosome. Typically, the number of variants depends on the length of the chromosome and for exome data the number of genes on the chromosome.
- Finally, `AgileROHFilterer` will state "*Filtering the VCF file*" indicating it is creating the results files. Since `AgileROHFinder` does not filter the variants, it just states "*Completed task*".

Unexpected results

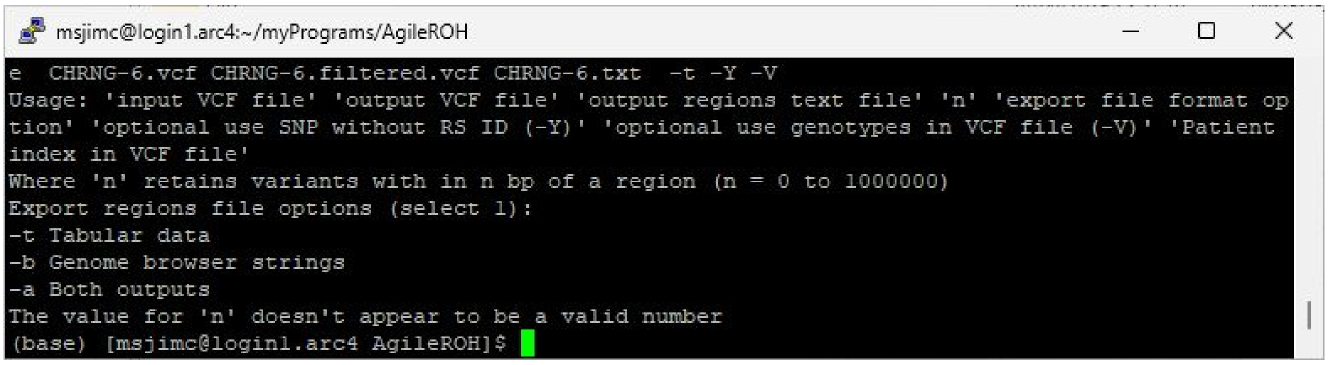
Both `AgileROHFilterer` and `AgileROHFinder` will try to process the arguments as best as possible and may not always detect an error. If the results are unexpected, check the messages described above to see if the command line has been written and processed correctly.

Failed analysis

Error in command line structure

Both `AgileROHFilterer` and `AgileROHFinder` will try to process the arguments as best as possible and may not always detect an error. A significant error that neither program will detect is the use of a data filename as an export filename, in which case the data file will be overwritten. Figure 3 shows the output from `AgileROHFilterer` if the 'n' value (number of bases by which a region is extended) is omitted or the arguments are in the wrong order. This message is very similar to the message displaced if the command line contains too few or too many arguments.

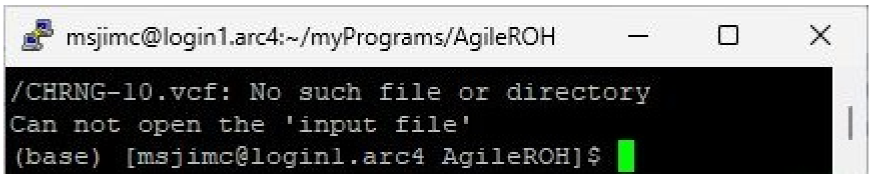
Figure 3: Wrong input file



```
e  CHRNG-6.vcf CHRNG-6.filtered.vcf CHRNG-6.txt  -t -Y -V
Usage: 'input VCF file' 'output VCF file' 'output regions text file' 'n' 'export file format op
tion' 'optional use SNP without RS ID (-Y)' 'optional use genotypes in VCF file (-V)' 'Patient
index in VCF file'
Where 'n' retains variants with in n bp of a region (n = 0 to 1000000)
Export regions file options (select 1):
-t Tabular data
-b Genome browser strings
-a Both outputs
The value for 'n' doesn't appear to be a valid number
(base) [msjimc@login1.arc4 AgileROH]$
```

Wrong input file name

Figure 4: Wrong input file



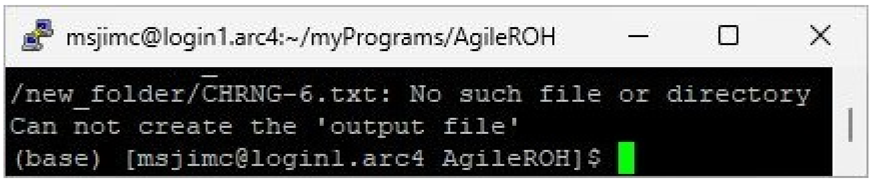
```
/CHRNG-10.vcf: No such file or directory
Can not open the 'input file'
(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 4: Feedback if the input file is incorrectly entered. A similar message will be displayed if the program can not open the file because it is open for editing in another program or you don't have permission to write

to the file.

Trying to export data to a folder that doesn't exist

Figure 5: Folder does not exist



msjimc@login1.arc4:~/myPrograms/AgileROH

```
/new_folder/CHRNG-6.txt: No such file or directory
Can not create the 'output file'
(base) [msjimc@login1.arc4 AgileROH]$
```

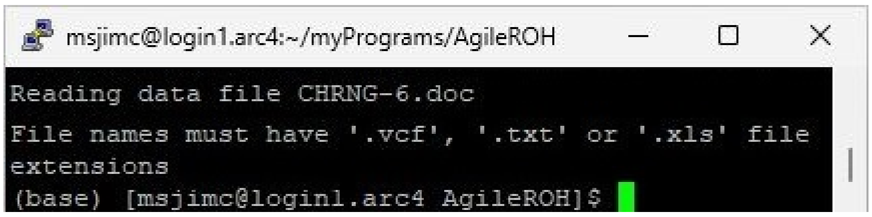
Figure 5: Feedback if the folder the export file is to be saved in does not exist. The programs can create results files, but will not create folders/directories.

File locked by another program or you don't have the necessary permissions

The error messages in Figures 4 and 5 may also be generated if you don't have permission to read/write to the files or folders. Similarly, if a file is open for editing in another program, access to it may be blocked resulting in a similar message.

The input data file's extension is not recognised

Figure 6: Unrecongised file extension



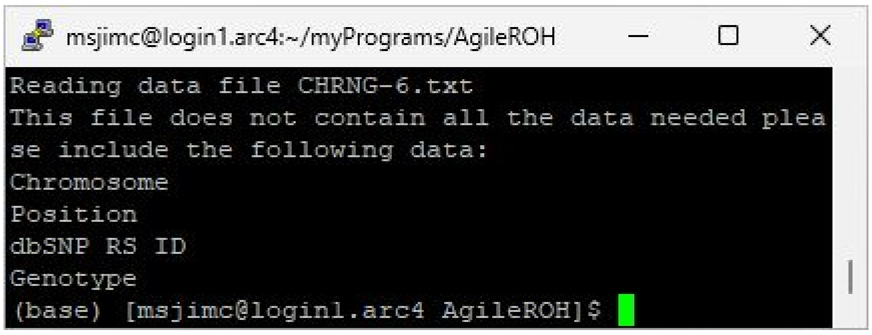
msjimc@login1.arc4:~/myPrograms/AgileROH

```
Reading data file CHRNG-6.doc
File names must have '.vcf', '.txt' or '.xls' file
extensions
(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 6: Feedback if the program does not recognise the file extension. While Linux itself does not use file extensions, these programs do use them to decide what type of data file is being used. If the file extension is not '.vcf', '.txt' or '.xls' the program will not process them.

The input data file's extension does not match the data type

Figure 7: Wrong file extension



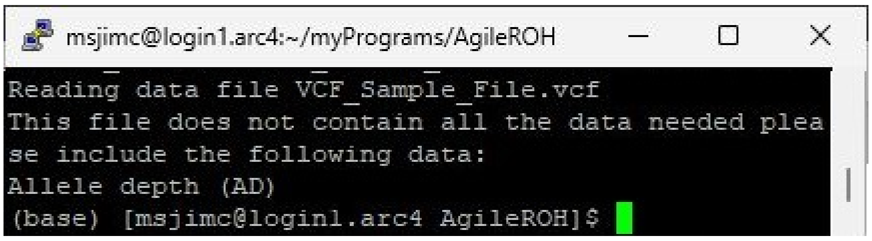
msjimc@login1.arc4:~/myPrograms/AgileROH

```
Reading data file CHRNG-6.txt
This file does not contain all the data needed plea
se include the following data:
Chromosome
Position
dbSNP RS ID
Genotype
(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 7: Feedback if the input file's extension does not match it's format. In this case the file is a vcf file, but its extension has been changed to txt. Consequently, the program tries to processed it as a microarray file and found that it does not contain the expected data fields/columns.

The input data file does not contain all the required data fields/columns

Figure 8: Missing data fields

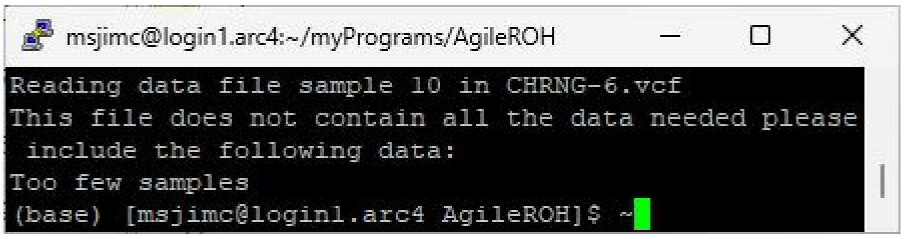


```
msjimc@login1.arc4:~/myPrograms/AgileROH
Reading data file VCF_Sample_File.vcf
This file does not contain all the data needed plea
se include the following data:
Allele depth (AD)
(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 8 shows the feedback if the input file does not contain the expected data fields/columns. In this case the vcf file is being processed without the '-V' option and while the file contains the total read depth value for each variant, it does not contain the read depths for each allele.

Sample index to high (VCF files only)

Figure 9: Missing data fields

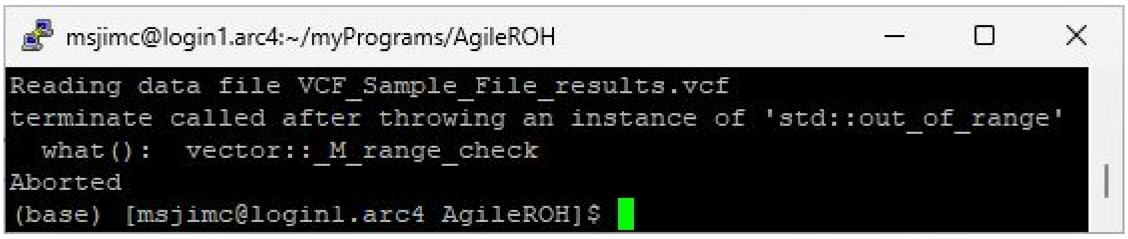


```
msjimc@login1.arc4:~/myPrograms/AgileROH
Reading data file sample 10 in CHRNG-6.vcf
This file does not contain all the data needed please
 include the following data:
Too few samples
(base) [msjimc@login1.arc4 AgileROH]$ ~
```

If a VCF file contains multiple samples it is possible to select which sample is processed by adding the samples index to the command line arguments. If the value is less than 1 it is ignored, if it is greater than the number of samples in the file the error message in Figure 9 is displayed.

The input data file's format is completely wrong

Figure 10: Wrong totally file format



```
msjimc@login1.arc4:~/myPrograms/AgileROH
Reading data file VCF_Sample_File_results.vcf
terminate called after throwing an instance of 'std::out_of_range'
  what():  vector::_M_range_check
Aborted
(base) [msjimc@login1.arc4 AgileROH]$
```

Figure 10: Feedback if the input file format is totally wrong and the program crashes reading it. In this case a results text file was given a vcf file extension and then entered as a vcf data file. The program has attempted to read data that does not exist and crashed. This will create a cryptic error message, if the problem persists after checking the files format, you may need to contact me.

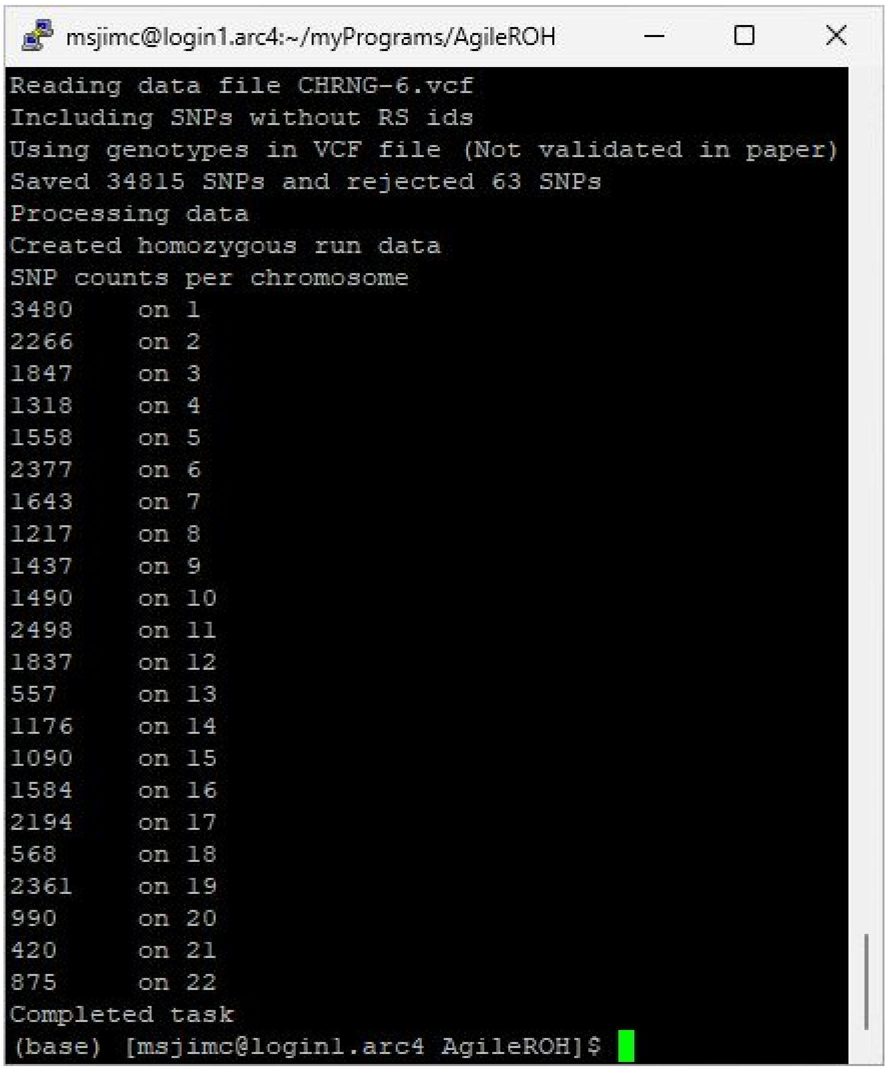
Note

There are many ways in which the input data or command line arguments lead the analysis to fail. While the commonest reasons have been listed above its possible that some combinations of input data format and command options will result in unexpected behaviour. In these cases always check that the file format matches the expected format as listed on the data format page.

Identifying regions in VCF files without read depth data

While many VCF files include read depth data for each variant, some VCF files do not. To allow the processing of this data, it is possible to instruct the programs to ignore read depth data and use the genotypes in the VCF file. In these cases the VCF file must contain the '**GT**' field and have have the genotypes declared as 0/0, 0/1, 1/0 or 1/1. Figure 11 shows the feedback of the analysis by `AgileROHFinder` with the optional **-V** used to set this behaviour.

Figure 11: Use of genotypes in VCF file


msjimc@login1.arc4:~/myPrograms/AgileROH

```
Reading data file CHRNG-6.vcf
Including SNPs without RS ids
Using genotypes in VCF file (Not validated in paper)
Saved 34815 SNPs and rejected 63 SNPs
Processing data
Created homozygous run data
SNP counts per chromosome
3480    on 1
2266    on 2
1847    on 3
1318    on 4
1558    on 5
2377    on 6
1643    on 7
1217    on 8
1437    on 9
1490    on 10
2498    on 11
1837    on 12
557     on 13
1176    on 14
1090    on 15
1584    on 16
2194    on 17
568     on 18
2361    on 19
990     on 20
420     on 21
875     on 22
Completed task
(base) [msjimc@login1.arc4 AgileROH]$
```

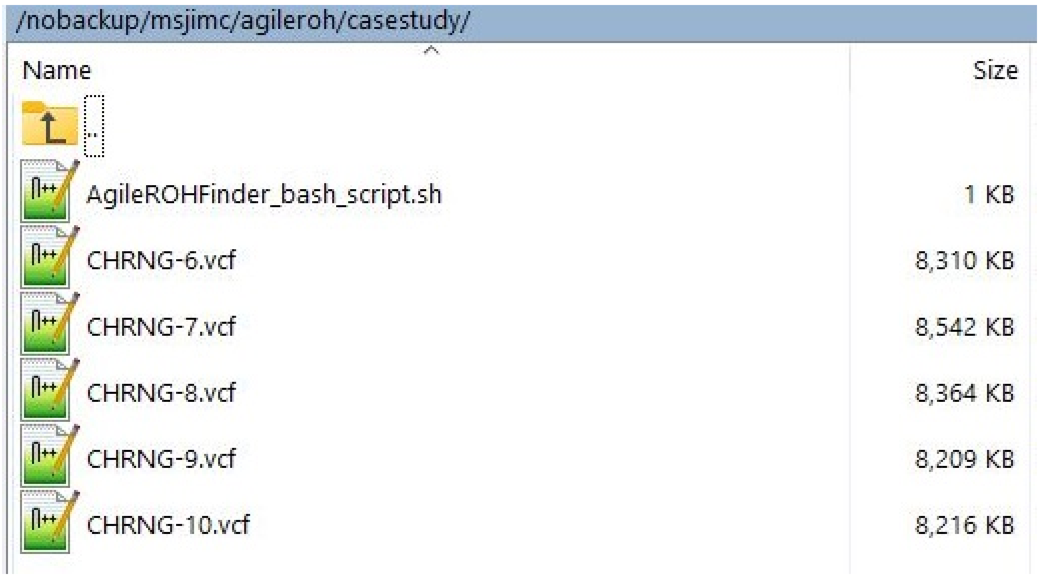
Important note: The results from this type of analysis were not investigated in the paper and so it is not supported. The quality of the results is highly likely to be dependant on the variant calling software and parameters used in its operation.

Case Study: CHRNG

This case study describes the analysis of the data in the "sample data file/CHRNG-family.zip" file on a remote Linux server using Putty and WinSCP. The analysis would be the same on a Windows computer except the paths would use back slashers ("\") rather then forward slashes ("/") and an absolute path would start with the drive letter (e.g. "C:\data\sample.vcf) instead of Linux's forward slash ("/data/sample.vcf").

Analysis: Find regions in folder of VCF files

Download and unzip the ***CHRNG-family.zip*** file, placing the VCF files in an empty folder (Figure 12). Download and save the ***AgileROHFinder_bash_script.sh*** to the server (in this example the file is also in the data folder).

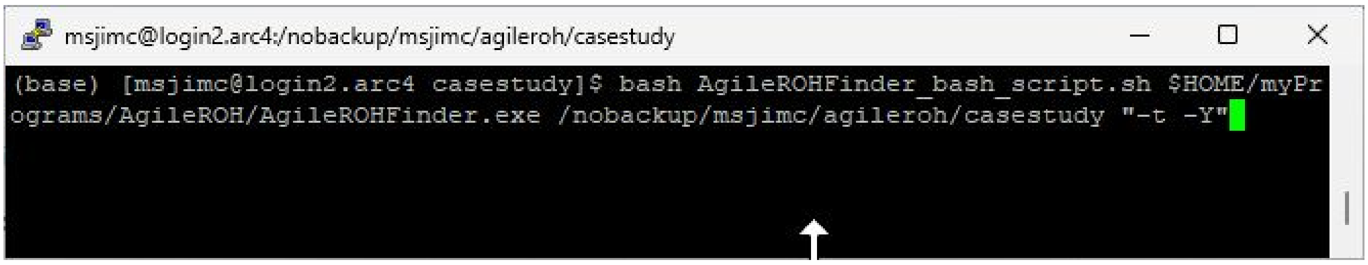


/nobackup/msjimc/agileroh/casestudy/

Name	Size
..	
AgileROHFinder_bash_script.sh	1 KB
CHRNG-6.vcf	8,310 KB
CHRNG-7.vcf	8,542 KB
CHRNG-8.vcf	8,364 KB
CHRNG-9.vcf	8,209 KB
CHRNG-10.vcf	8,216 KB

Figure 12

Open a SSH terminal and navigate to the data folder and issue the analysis command (Figure 13).



msjimc@login2.arc4:/nobackup/msjimc/agileroh/casestudy

```
(base) [msjimc@login2.arc4 casestudy]$ bash AgileROHFinder_bash_script.sh $HOME/myPr
ograms/AgileROH/AgileROHFinder.exe /nobackup/msjimc/agileroh/casestudy "-t -Y"
```

Figure 13

In this example the command is (all on one line):

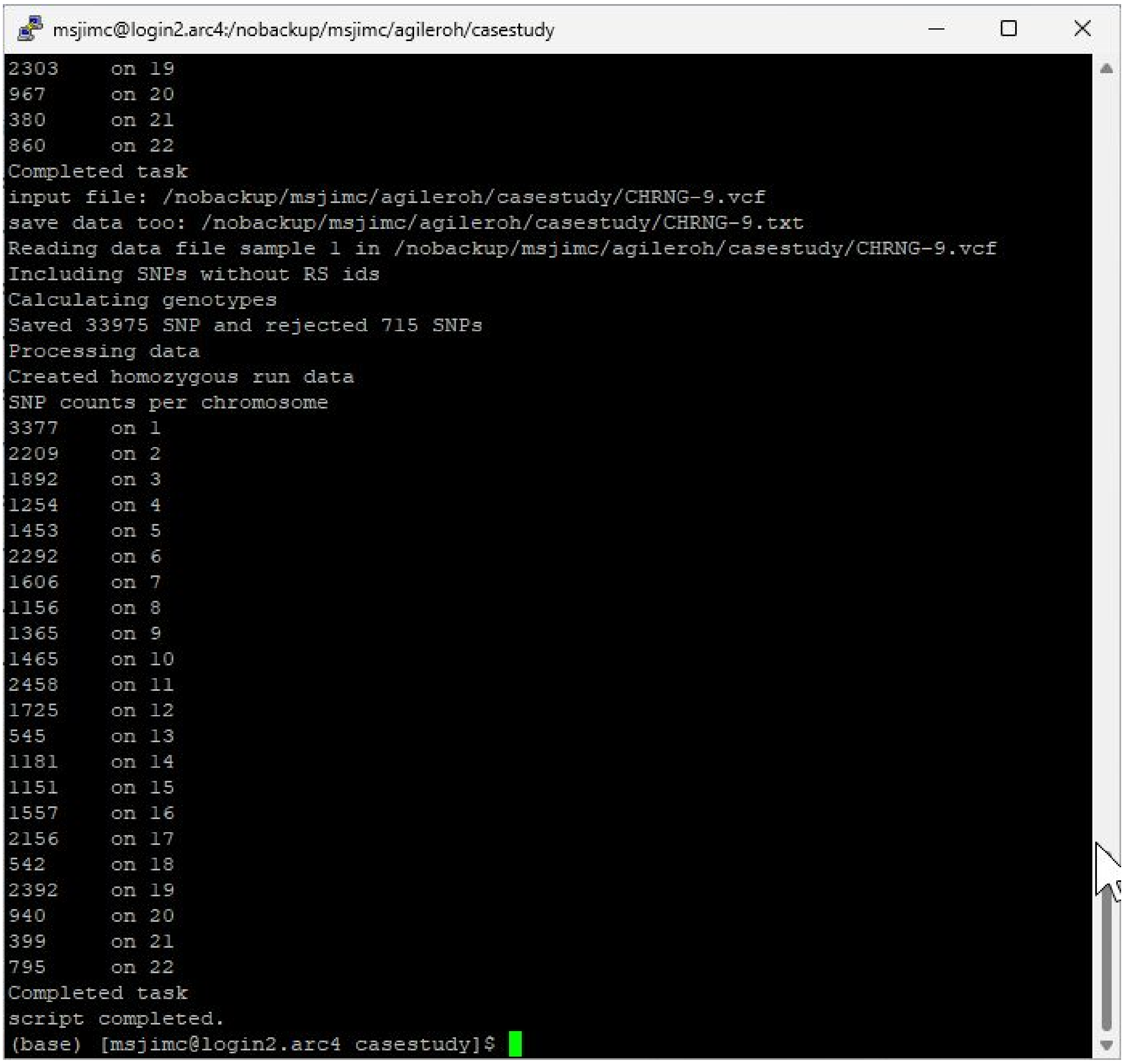
> bash AgileROHFinder_bash_script.sh $HOME/myPrograms/AgileROH/AgileROHFinder.exe /nobackup/msjimc/agileroh/casestudy "-t -Y"

where

Parameter	Note
bash	Instructs the terminal run the command as a bash script
AgileROHFinder_bash_script.sh	Name of script with 'relative' path
$HOME/myPrograms/AgileROH/AgileROHFinder.exe	Location of the AgileROHFinder.exe program, in this case it is in a sub-folder of my home directory
/nobackup/msjimc/agileroh/casestudy	Name and path to data folder

Parameter	Note
"***-t -Y***"	Formatting parameter with the optional ***-Y*** parameter. Note the ***-t*** and ***-Y*** are in speech marks, if they are omitted only the -t parameter will be used

As the script runs, `AgileROHFinder.exe` is instructed to analyses each data file in turn and produces a series of outputs as described above (Figure 14).

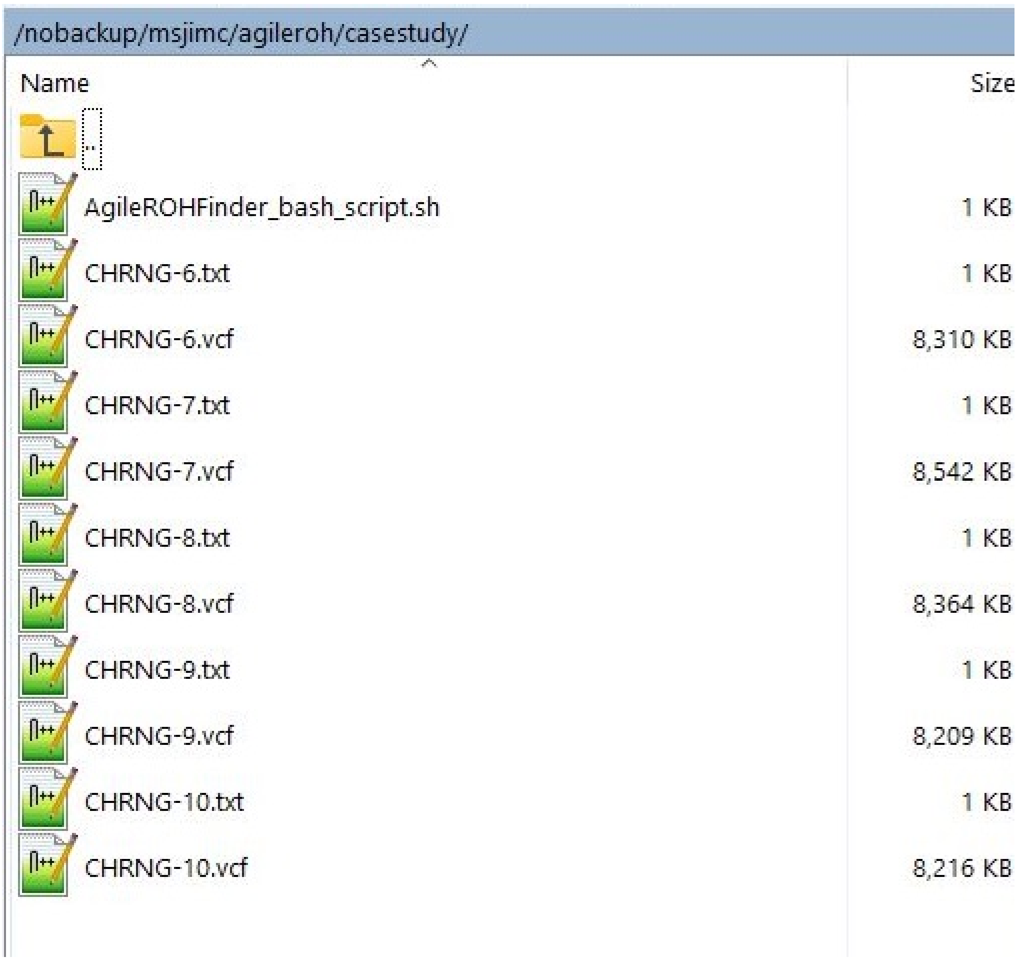


msjimc@login2.arc4:/nobackup/msjimc/agileroh/casestudy

```
2303    on 19
967     on 20
380     on 21
860     on 22
Completed task
input file: /nobackup/msjimc/agileroh/casestudy/CHRNG-9.vcf
save data too: /nobackup/msjimc/agileroh/casestudy/CHRNG-9.txt
Reading data file sample 1 in /nobackup/msjimc/agileroh/casestudy/CHRNG-9.vcf
Including SNPs without RS ids
Calculating genotypes
Saved 33975 SNP and rejected 715 SNPs
Processing data
Created homozygous run data
SNP counts per chromosome
3377    on 1
2209    on 2
1892    on 3
1254    on 4
1453    on 5
2292    on 6
1606    on 7
1156    on 8
1365    on 9
1465    on 10
2458    on 11
1725    on 12
545     on 13
1181    on 14
1151    on 15
1557    on 16
2156    on 17
542     on 18
2392    on 19
940     on 20
399     on 21
795     on 22
Completed task
script completed.
(base) [msjimc@login2.arc4 casestudy]$
```

Figure 14

When completed, each VCF file will have a matched text file containing its autozygous regions (Figure 15). These files can then be manually viewed or used to filter the variants in a subsequent downstream filtering step in the patient data analysis pipeline.



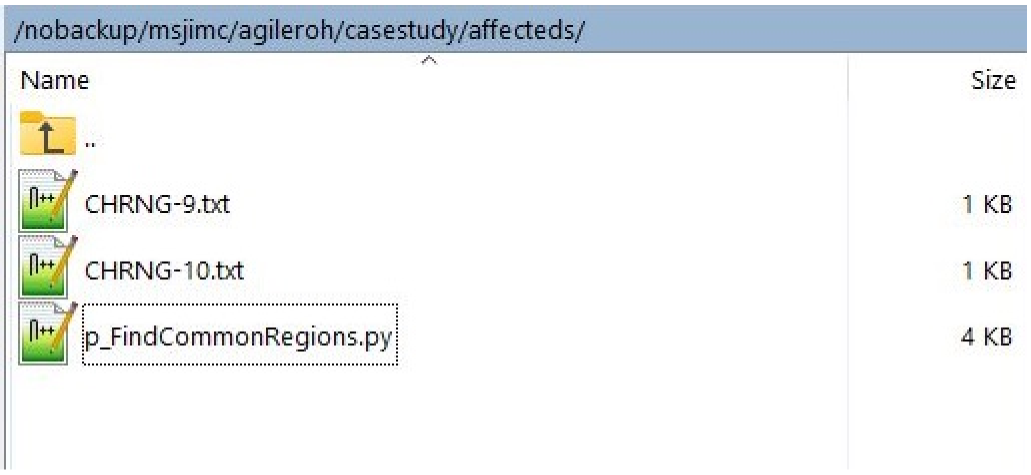
/nobackup/msjimc/agileroh/casestudy/

Name	Size
..	
AgileROHFinder_bash_script.sh	1 KB
CHRNG-6.txt	1 KB
CHRNG-6.vcf	8,310 KB
CHRNG-7.txt	1 KB
CHRNG-7.vcf	8,542 KB
CHRNG-8.txt	1 KB
CHRNG-8.vcf	8,364 KB
CHRNG-9.txt	1 KB
CHRNG-9.vcf	8,209 KB
CHRNG-10.txt	1 KB
CHRNG-10.vcf	8,216 KB

Figure 15

Analysis: Find common regions in affected samples

To view the regions common to the affected individuals, more the affected samples' regions files to an empty folder and download the ***p_FindCommonRegions.py*** python script (Figure 16).

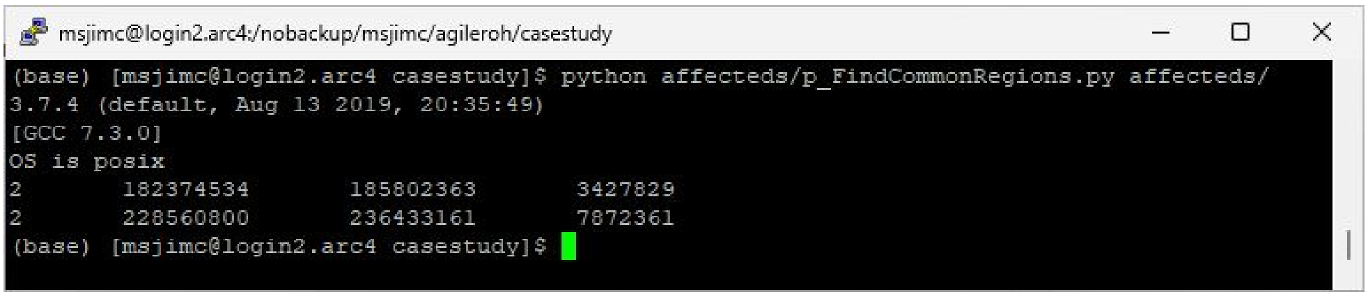


/nobackup/msjimc/agileroh/casestudy/affecteds/

Name	Size
..	
CHRNG-9.txt	1 KB
CHRNG-10.txt	1 KB
p_FindCommonRegions.py	4 KB

Figure 16

From the terminal issue the final command (Figure 17):



```
(base) [msjimc@login2.arc4 casestudy]$ python affecteds/p_FindCommonRegions.py affecteds/
3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0]
OS is posix
2       182374534       185802363       3427829
2       228560800       236433161       7872361
(base) [msjimc@login2.arc4 casestudy]$
```

Figure 17

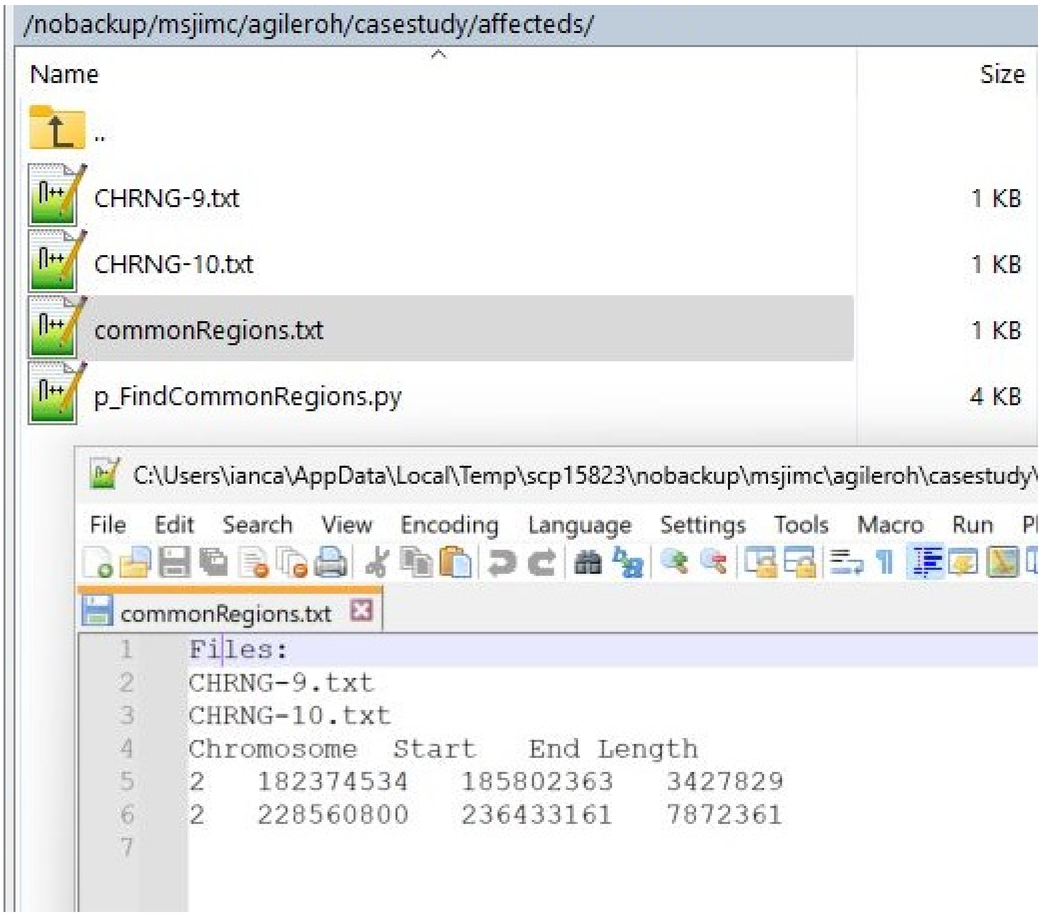
In this example the command is:

> python affecteds/p_FindCommonRegions.py affecteds/

where:

Parameter	Note
python	Instructs the terminal to process a python script
affecteds/p_FindCommonRegions.py	Name with path of the ***p_FindCommonRegions.py*** script
affecteds/	The name with path to the folder of regions files

As the python script runs, it first displays the version and type of OS that the server is using. This allows the script to determine the type of filesystem and whether to use "\" or "/".
The subsequent lines display the location and length of any common autozygous regions (Figure 17). This data is also saved a file called ***commonRegions.txt*** which also contains the list of file names used in the analysis (Figure 18).


/nobackup/msjimc/agileroh/casestudy/affecteds/
Name
Size
..
CHRNG-9.txt 1 KB
CHRNG-10.txt 1 KB
commonRegions.txt 1 KB
p_FindCommonRegions.py 4 KB
C:\Users\ianca\AppData\Local\Temp\scp15823\nobackup\msjimc\agileroh\casestudy\
File Edit Search View Encoding Language Settings Tools Macro Run Pl
commonRegions.txt

```
Files:
CHRNG-9.txt
CHRNG-10.txt
Chromosome  Start   End Length
2   182374534   185802363   3427829
2   228560800   236433161   7872361
```

Figure 18