

# 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 older Affymetrix 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 only analyse one patient per file, if a file contains data on multiple individuals only one person will be analysed.

## Creating the programs

The source code can be compiled on both Linux and Windows computers as described [here](#).

## Prebuild programs

Both programs have been prebuilt for Linux and Windows and placed in the Program folder. Select the appropriate OS version 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.

## 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 connected 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 very similar commands, the structure of the commands to run [AgileROHFinder](#) and [AgileROHFilterer](#) are shown below and described in greater detail in the table.

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
```

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	Name of a file to save the filtered variant data too.	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.
/data/out.txt	The name with location of the file to save the list of autozygous regions too	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.
Any whole positive number	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	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

Command fragment	Description	Note
Process all variants: <b>-Y</b> or <b>-N</b>	By default only variants with an RS ID are processed, if <b>-Y</b> is set then all SNPs (with one alternative allele) will be used	Optional

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

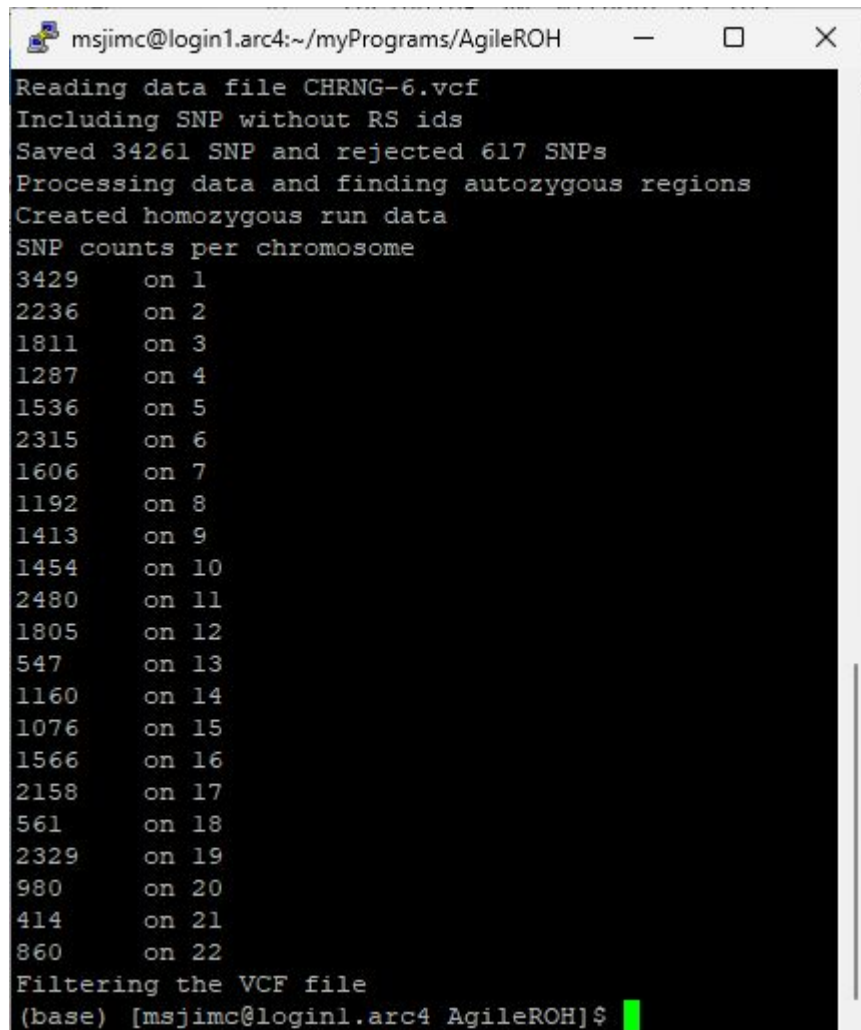
Genome browser
chr2:25656880-29092679
chr2:179421694-180835792
chr2:182374534-189875421
chr11:48367050-55111584
chr17:21318629-26691321
```

Feedback

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

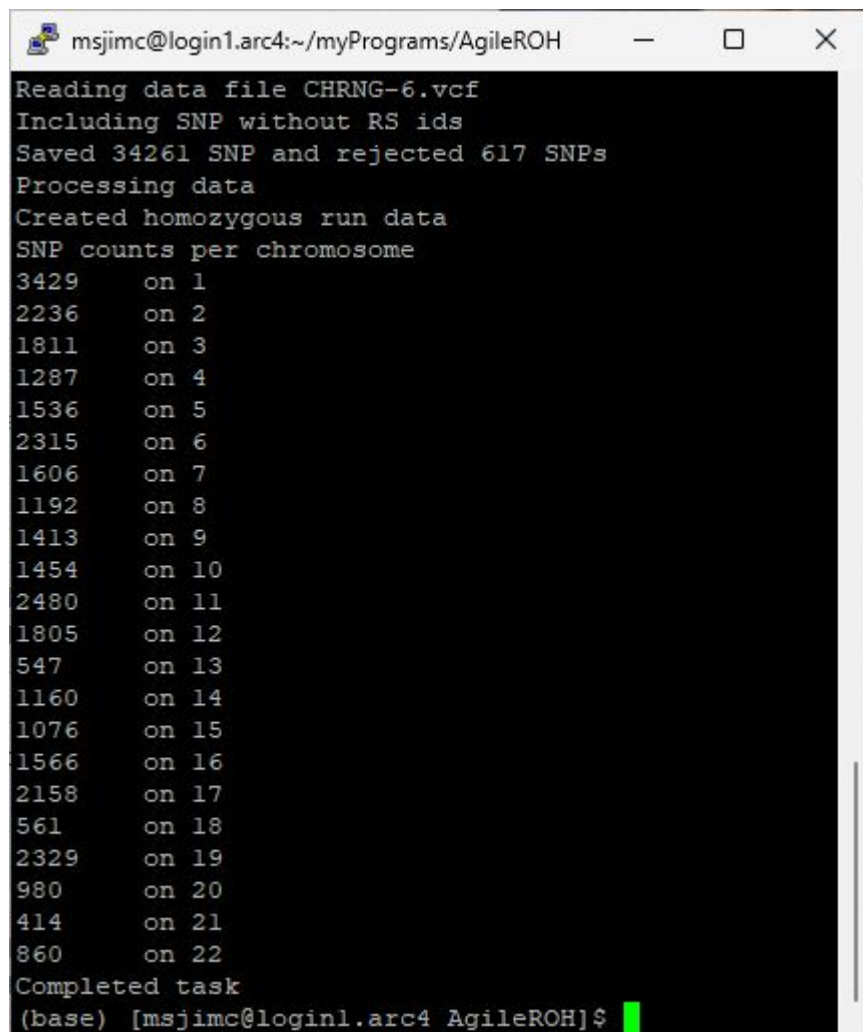
A successful analysis:

Figure 1: AgileROHFilterer

A terminal window titled 'msjmc@login1.arc4:~/myPrograms/AgileROH' with standard window controls. The terminal displays the output of the AgileROH program. It starts by reading a VCF file, then processes SNPs, saving 34261 and rejecting 617. It then finds autozygous regions and creates homozygous run data. A table follows showing SNP counts per chromosome from 1 to 22. The process ends with filtering the VCF file and a shell prompt.

```
msjmc@login1.arc4:~/myPrograms/AgileROH
Reading data file CHRNG-6.vcf
Including SNP without RS ids
Saved 34261 SNP and rejected 617 SNPs
Processing data and finding autozygous regions
Created homozygous run data
SNP counts per chromosome
3429    on 1
2236    on 2
1811    on 3
1287    on 4
1536    on 5
2315    on 6
1606    on 7
1192    on 8
1413    on 9
1454    on 10
2480    on 11
1805    on 12
547     on 13
1160    on 14
1076    on 15
1566    on 16
2158    on 17
561     on 18
2329    on 19
980     on 20
414     on 21
860     on 22
Filtering the VCF file
(base) [msjmc@login1.arc4 AgileROH]$
```

Figure 2: AgileROHFinder



```
msjmc@login1.arc4:~/myPrograms/AgileROH
Reading data file CHRNG-6.vcf
Including SNP without RS ids
Saved 34261 SNP and rejected 617 SNPs
Processing data
Created homozygous run data
SNP counts per chromosome
3429    on 1
2236    on 2
1811    on 3
1287    on 4
1536    on 5
2315    on 6
1606    on 7
1192    on 8
1413    on 9
1454    on 10
2480    on 11
1805    on 12
547     on 13
1160    on 14
1076    on 15
1566    on 16
2158    on 17
561     on 18
2329    on 19
980     on 20
414     on 21
860     on 22
Completed task
(base) [msjmc@login1.arc4 AgileROH]$
```

Figure 1

Figures 1 and 2 show a typical status report of the analysis of a exome vcf file by **AgileROHFilterer** and **AgileROHFinder** respectively.

### Output description:

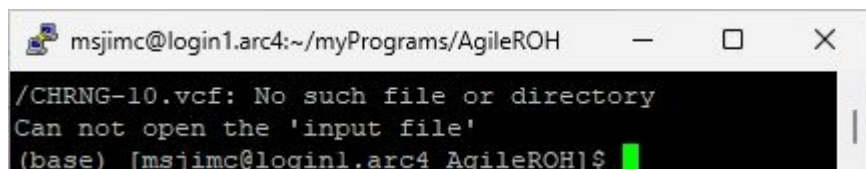
- Initially, the program states which file is being process.
- Next it states whether it will process variants without an RS ID. "*Including SNP without RS ids*" indicates it will process all variants, while "*Ignoring SNP without RS IDs*" indicates unnamed variants will be excluded.
- Once the file 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 being 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*".

## Failed analysis

### Wrong input file name

**Figure 3: Wrong input file**

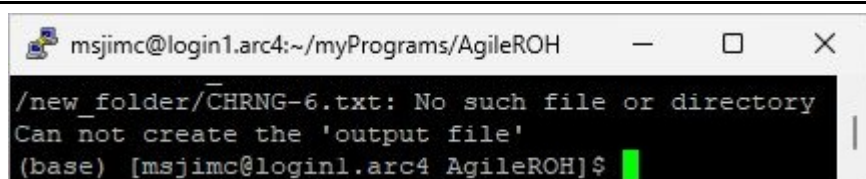


```
msjmc@login1.arc4:~/myPrograms/AgileROH
/CHRNA-10.vcf: No such file or directory
Can not open the 'input file'
(base) [msjmc@login1.arc4 AgileROH]$
```

Figure 3: 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 4: Folder does not exist**

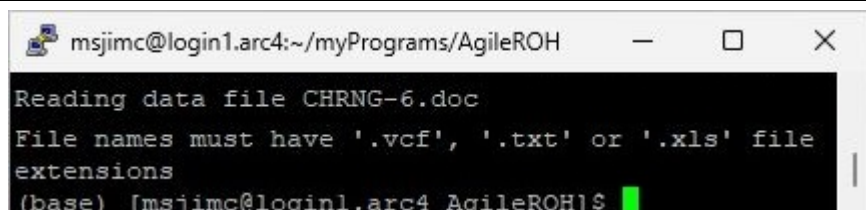


```
msjmc@login1.arc4:~/myPrograms/AgileROH
/new_folder/CHRNA-6.txt: No such file or directory
Can not create the 'output file'
(base) [msjmc@login1.arc4 AgileROH]$
```

Figure 4: 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.

### The input data file's extension is not recognised

**Figure 5: Unrecognised file extension**

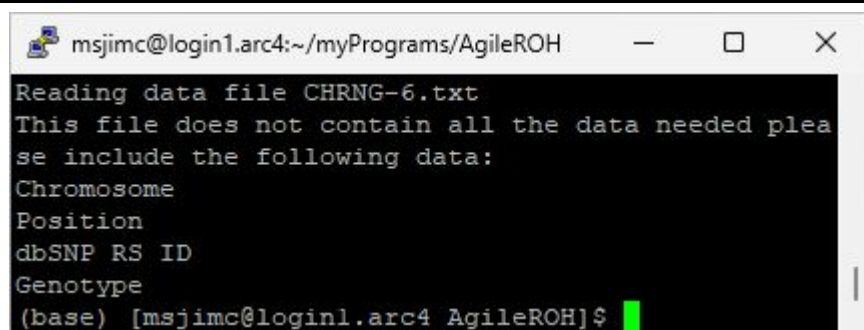


```
msjmc@login1.arc4:~/myPrograms/AgileROH
Reading data file CHRNA-6.doc
File names must have '.vcf', '.txt' or '.xls' file
extensions
(base) [msjmc@login1.arc4 AgileROH]$
```

Figure 5: Feedback if the input file 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

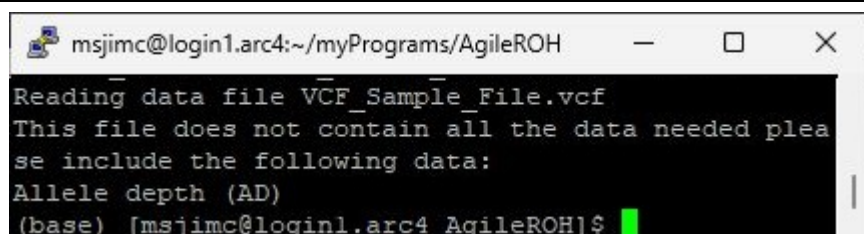
**Figure 6: Wrong file extension**

**Figure 6: Wrong file extension**

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

Figure 6: Feedback if the input file's extension does not match its format. In this case the file is a vcf file, but its extension has been changed to txt. Consequently, the program tries to process it as a microarray file and found that it does not contain the expected data fields/columns.

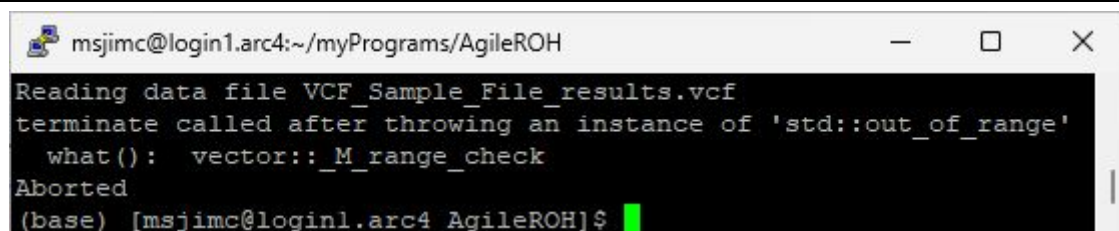
The input data file does not contain all the required data fields/columns

**Figure 7: Missing data fields**

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

Figure 7 shows the program feedback if input file does not contain the expected data fields/columns. In this case the vcf file contains the total read depth value for each variant, but not the read depths for each allele.

The input data file's format is completely wrong

**Figure 8: Wrong totally file format**

```
msjmc@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) [msjmc@login1.arc4 AgileROH]$
```

Figure 8: 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 attempted to read data that does not exist and crashed. This will create a cryptic error message, if the problem persists after checking the file's 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 it's possible that some combinations of input data format and

command option will result in unexpected behaviour. In these case always check that the file format matches the expected format as listed On the [data format page](#).