Title

BrowseVCF: a web-based application and workflow to quickly prioritise disease-causative variants in VCF files.

Authors

Silvia Salatino[1,*], Varun Ramraj[1], Stefano Lise[1,2].

Author information

[1] The Wellcome Trust Centre for Human Genetics, Roosvelt Drive, Oxford OX3 7BN, UK

[2] Current address: The Institute of Cancer Research, 15 Cotswold Road, Sutton, Surrey SM2 5NG, UK

* Corresponding author

Abstract

As sequencing cost continues to decrease with fast advancing Next Generation Sequencing (NGS) technologies, variant discovery is becoming a more affordable and popular analysis method among research laboratories. Following variant calling and annotation, accurate variant filtering is a crucial step to extract meaningful biological information from sequencing data and to investigate disease etiology. However, the standard variant call file format (VCF) used to store this valuable information is not easy to handle without bioinformatics skills, thus preventing investigators from directly analysing their data. Here, we present BrowseVCF, an easy-to-use stand-alone software that enables researchers to browse, query and filter millions of variants in a few seconds. Top features include the possibility to store intermediate search results, to query user-defined gene lists, to group samples for family or tumour/normal studies, to download a report of the filters applied, and to export the filtered variants in spreadsheet format. Additionally, BrowseVCF is suitable for any DNA variant analysis (exome, whole-genome and targeted sequencing), can be used also for non-diploid genomes, and is able to discriminate between Single Nucleotide Polymorphisms (SNPs), Insertions/Deletions (InDels), and Multiple Nucleotide Polymorphisms (MNPs). Thanks to its portable implementation, BrowseVCF can be used either on personal computers or as part of automated analysis pipelines. The software can be initialised with a few clicks on any operating system without any special administrative or installation permissions, and is freely available for download from https://github.com/BSGOxford/BrowseVCF.

Keywords

VCF; variant filtering; variant analysis; exome sequencing; whole-genome sequencing; prioritisation.

Introduction

Recent developments of NGS technologies have led to a dramatic reduction in sequencing costs which, in turn, made DNA sequencing analyses accessible to small and medium-sized laboratories. As a consequence, increasing amounts of sequencing data need to be analysed quickly and require specific bioinformatics expertise. Whilst large institutes might be able to rely on support from dedicated bioinformatics core units, wet-lab researchers from minor centres struggle to find easy-to-use tools for variant analysis. The variant call format (VCF), originally developed for the 1000 Genomes Project, has become the standard for storing DNA variants together with rich annotations [1], and has led to the development of a variety of command-line analysis tools (for instance, VCFtools [1] or GEMINI [2]), although a graphical interface is more accessible to non-bioinformaticians. Recently, efforts have been made towards developing more user-friendly graphical software; examples include the commercial suites Ingenuity [3], Alamut [4], GoldenHelix SNP [5] and VariantStudio [6], as well as the open-source packages SNVerGUI [7], database.bio [8], gNOME [9] and BiERapp [10]. However, the major disadvantage common to these tools is that they all strongly depend on a well-defined set of annotations, which limits the user to a restricted number of pre-defined features. Instead, researchers might be interested in keeping their personal or in-house annotations for downstream analysis, rather than discarding them. To our knowledge, only one recently-published software (VCF-Miner) addresses this problem by skipping the annotation step and focusing on the filtering part [11]. However, it has a lengthy installation process, VCF file load time, and does not allow the user to query variants which are annotated with a given set of genes of interest.

BrowseVCF overcomes these limitations and provides a fast, portable, and simple interactive analysis tool to non-bioinformatician researchers. Moreover, our software has a new feature that allows users to perform keyword-based searches to identify, for instance, specific consequence terms or associated disorder states.

By applying filters sequentially to the input set of variants, BrowseVCF empowers researchers to narrow down the amount of putative variants to a small number of candidate disease-causing alleles.

Methods

VCF file loading and pre-processing

The VCF file format is well-defined [1]. It consists of a header section, containing an arbitrary number of meta-information lines that start with the symbol '#', and of a data section, containing one line per variant, split in eight mandatory columns: chromosome (CHROM), 1-based starting position of the variant (POS), unique identifier, if existing (ID), reference allele in the genome (REF), alternative allele(s) of the variant (ALT), phred-scaled quality score (QUAL), flag for passed/failed control checks (FILTER) and variant-specific annotations (INFO), which can be an unrestricted number of either flags (present/absent) or key-value pairs. VCF files containing one or more samples also include a ninth column (FORMAT), used to define the information enclosed in each subsequent column, and a genotype column for each sample, regarding the allele combination, the genotype read depth and other metrics.

The loading step within BrowseVCF is a very straightforward process. The user selects a VCF file from his personal computer through a pop-up dialog box, which is then pre-processed. BrowseVCF can accept both uncompressed and compressed (*.gz) VCF file types. The software identifies every annotation field present in the input VCF file and presents this list to the user, who can then select the fields of interest that will be used to filter the variants (Fig. 1). Additional fields can be selected at any time, if needed.

The “Wormtable” Python module [12] was chosen over other data storage systems for its efficient use of disk space and for the speed of its queries. However, as one wormtable is created per chosen field, selecting fewer fields will reduce the time it takes for BrowseVCF to process the file. The pre-processing step is also required to convert missing values ('.') to '-1' and to convert CSQ annotations (in case the file was annotated with Variant

Effect Predictor [13]) into a Wormtable-compatible format.

Variant prioritization by sequential filtering

Once the file is pre-processed, the user can start to filter variants according to five different query types (Fig. 2). The first filter lets the user query the data based on a given field of interest. Based on the type of annotation, different operators can be chosen for a specific cutoff/keyword (“greater_than”, “less_than”, “equal_to” or “contains_keyword”) and the user can choose whether to include or exclude variants with missing values. Moreover, the “contains_keyword” option allows to look for multiple keywords within the same field, which proves very useful when the objective is to keep, for example, variants labelled with different consequence or disease tags. The second filter extracts variants with a given genotype in one, some or all samples; this feature is particularly useful in the study of inherited and *de novo* syndromes within a family or close relatives. BrowseVCF is able to process both diploid and non-diploid genomes alike. The third filter can retrieve all variants located within a given region of interest; in this case, the user is prompted to specify chromosome, start position and end position. This option is particularly useful to integrate other sources of information, e.g. regions identified by linkage analysis or targeted for capture. The fourth filter is able to extract either SNPs, InDels, or MNPs. Finally, the fifth query allows the user to either upload a text file containing one gene per line or to specify a comma-separated list of genes and retrieves all variants that are either associated (positive query) or not (negative query) to any of those entries within a specific annotation field. Filters are applied sequentially, so that a new filter is applied only to the variants that have successfully passed the previous filter.

Exporting results from the graphical interface

BrowseVCF comes with a user-friendly graphical interface (GUI) that allows researchers to load, filter, track, and export variants and annotations. Figure 2 displays the filtering interface. The right panel of the GUI displays variants in a tabular form, whereas the left panel keeps track of the filters applied and of the number of variants resulting from each query. At any point the user can download the query history for record-keeping. Also, the output of each query is automatically stored and can be deleted at any time from the “Filter History” section in the left panel. The right panel shows a preview of the top 100 variants and all variants can be exported at any time as a tab-separated text file, compatible with spreadsheet programs like Microsoft Excel.

Technical details

BrowseVCF is free software under the GNU General Public License version 3 [14]. It consists of a back-end service of Python scripts and a web server that runs locally, and of a front end built in JavaScript, CSS, and HTML5, leveraging popular cross-platform libraries such as AngularJS and Bootstrap. In order to perform fast and memory-efficient queries, all filtering operations are executed using Wormtable, a recently-developed Python module optimised to store genomic data in a compact binary tabular format through Berkeley DB [12]. To enable cross-platform usage, especially on Windows, Wormtable was altered and optimised slightly. Every operation is carried out locally on the user's computer, using the default browser; BrowseVCF is supported by all modern browsers, including Firefox and Chrome. The Windows version of BrowseVCF is shipped along with a lightweight distribution of WinPython, a bundled environment pre-configured to run directly on Windows. This enables the Windows user to simply double-click on an executable file without needing administrative permissions to install Python and necessary custom modules, which greatly improves the accessibility and ease-of-use of our software. However, since the tool can be configured to run with any web server back-end, it can also be installed site-wide within an organisation, thereby enabling shared use.

Command-line version of BrowseVCF

BrowseVCF can also be used from command-line as a set of one-off queries, or as part of automated analysis pipelines. Each Python script is independent from the others and the list of required arguments can be displayed by typing the script name followed by "--help". Some of the scripts are capable of parallel execution by specifying the number of cores at invocation, allowing BrowseVCF to play nicely within a high-performance computing (HPC) environment.

Results

Benchmarks

We checked the performances of BrowseVCF on two different file types: an exome trio (proband, mother, father) and the whole-genome v2.18 of sample NA12878 generated by the "Genome in a Bottle Consortium" [15]. As shown in Table 1, the amount of time required to pre-process the VCF file is below 2 minutes for standard exome data and the first query takes usually less than 30 seconds. Since each subsequent query uses the filtered output of the previous one, execution time gets shorter. For whole-genome data, pre-processing and index creation time is longer than for an exome VCF (as expected). Timings were collected using a single core and differed between the Windows and the GNU/Linux machines. Specifying a higher number of CPUs to use in step 2 speeds up the process of creating the wormtables for the fields of interest.

Table 1. Performances of BrowseVCF on exome and whole-genome data. Pre-processing times vary between operating system due to implementation differences intrinsic to Python.

VCF file	Size (MB)	Variants	Samples	Step 1: Pre-processing time[1]		Step 2: Index creation time[2]		Step 3: Filtering time[3]	
				PC1	PC2	PC1	PC2	PC1	PC2
Exome_trio.vcf.gz	22	100014	3	0min 22sec	0min 18sec	1min 2sec	0min 35sec	0min 28sec	0min 14sec
GIAB_v2.18.vcf.gz	313	2915731	1	6min 29sec	5min 8sec	37min 53sec	9min 11sec	13min 9sec	3min 38sec

[1] Step required to convert the input VCF file to a format accepted by Wormtable
[2] Wormtables generated for the following fields: CHROM+POS, ID, REF+ALT, QUAL, FILTER
[3] Query executed on the FILTER field, keeping only PASS variants

Note: PC1 is a Windows machine with 8 GB RAM; PC2 is a GNU/Linux machine with 8 GB RAM.
All operations were performed using only 1 CPU.

Discussion

The crucial step in variant analysis studies is an accurate prioritisation of the identified mutations, in order to pinpoint a small number of putative candidates which might have caused a given disease or phenotype. BrowseVCF is a fast and intuitive interactive software that enables researchers to browse and query their data without the need of any bioinformatics expertise. The major advantage of BrowseVCF with respect to similar tools is in its fast implementation, which turns out to be particularly useful when analysing millions of variants from whole-genome sequencing data. Additionally, BrowseVCF is very straightforward to use on any operating system as it does not require administrator privileges to install and run. However, BrowseVCF shows a propensity for shorter execution times on GNU/Linux machines over Windows machines of similar specification due to operating-system specific differences in Python.

Another limitation of existing tools is the need to load the entire VCF file, even though no more than a dozen annotations may typically be used for the analysis. Instead, BrowseVCF allows the user to choose which annotations to process before running the filters and, if needed, to go back and load additional fields for further filtering steps. Considering that the number of samples and annotations in VCF files are the main drivers of the load time, being able to restrict the analysis only to the subset of truly relevant fields saves both computational time and disk space.

In addition, our software can be used both interactively and as part of computational pipelines. Also, the ability to perform all the analysis locally is a considerable advantage when dealing with clinical sensitive files. It is understood in the clinical world that web-based interfaces are easier to operate without the need for command-line scripts, but pose a security risk when dealing with sensitive patient data [16]. We designed BrowseVCF with flexibility in mind, as it can run on a personal laptop without administrative privileges, as part of a HPC environment, and in cases where patient data are available to an organisation behind its firewall, BrowseVCF can be deployed centrally as a web service. In cases where data sensitivity is less of an issue, BrowseVCF can also be deployed as a public online web service. Future releases of BrowseVCF will allow saving and resuming sessions, and global configuration options. In conclusion, BrowseVCF can significantly improve efficiency in navigating large data sets to find candidate disease variants, thereby streamlining the clinical research process.

Disclosure declaration

The authors declare that there is no conflict of interest associated with this manuscript.

Author contributions

S.S. and S.L. designed the project. S.S. and V.R. implemented the code. S.S. and V.R. wrote the manuscript. All authors have read and approved the manuscript.

Funding

This work was funded by a Wellcome Trust Core Award (090532/Z/09/Z), supporting the Core facilities at The Wellcome Trust Centre for Human Genetics, including the Oxford Genomics Centre.

Acknowledgements

We thank Jerome Kelleher for significant help with implementation of new features in Wormtable and Kate Thomson, John Taylor, and Alistair Pagnamenta for useful discussion and feedback on BrowseVCF.

References

[1] Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R; 1000 Genomes Project Analysis Group. The variant call format and VCFtools. *Bioinformatics*. 2011 Aug 1;27(15):2156-8.

[2] Paila U, Chapman BA, Kirchner R, Quinlan AR. GEMINI: integrative exploration of genetic variation and genome annotations. *PLoS Comput Biol*. 2013;9(7):e1003153.

[3] http://www.ingenuity.com/

[4] http://www.interactive-biosoftware.com/alamut-visual/

[5] http://goldenhelix.com/SNP_Variation/index.html

[6] http://www.illumina.com/informatics/research/biological-data-interpretation/variantstudio.html

[7] Wang W, Hu W, Hou F, Hu P, Wei Z. SNVerGUI: a desktop tool for variant analysis of next-generation sequencing data. *J Med Genet*. 2012 Dec;49(12):753-5.

[8] Ou M, Ma R, Cheung J, Lo K, Yee P, Luo T, Chan TL, Au CH, Kwong A, Luo R, Lam TW. database.bio: a web application for interpreting human variations. *Bioinformatics*. 2015 Aug 26. pii: btv500.

[9] Lee IH, Lee K, Hsing M, Choe Y, Park JH, Kim SH, Bohn JM, Neu MB, Hwang KB, Green RC, Kohane IS, Kong SW. Prioritizing disease-linked variants, genes, and pathways with an interactive whole-genome analysis pipeline. *Hum Mutat*. 2014 May;35(5):537-47.

[10] Alemán A, Garcia-Garcia F, Salavert F, Medina I, Dopazo J. A web-based interactive framework to assist in the prioritization of disease candidate genes in whole-exome sequencing studies. *Nucleic Acids Res*. 2014 Jul;42(Web Server issue):W88-93.

[11] Hart SN, Duffy P, Quest DJ, Hossain A, Meiners MA, Kocher JP. VCF-Miner: GUI-based application for mining variants and annotations stored in VCF files. *Brief Bioinform*. 2015 Jul 25. pii: bbv051.

[12] Kelleher J, Ness RW, Halligan DL. Processing genome scale tabular data with wormtable. *BMC Bioinformatics*. 2013 Dec 5;14:356.

[13] http://www.ensembl.org/info/docs/tools/vep/index.html

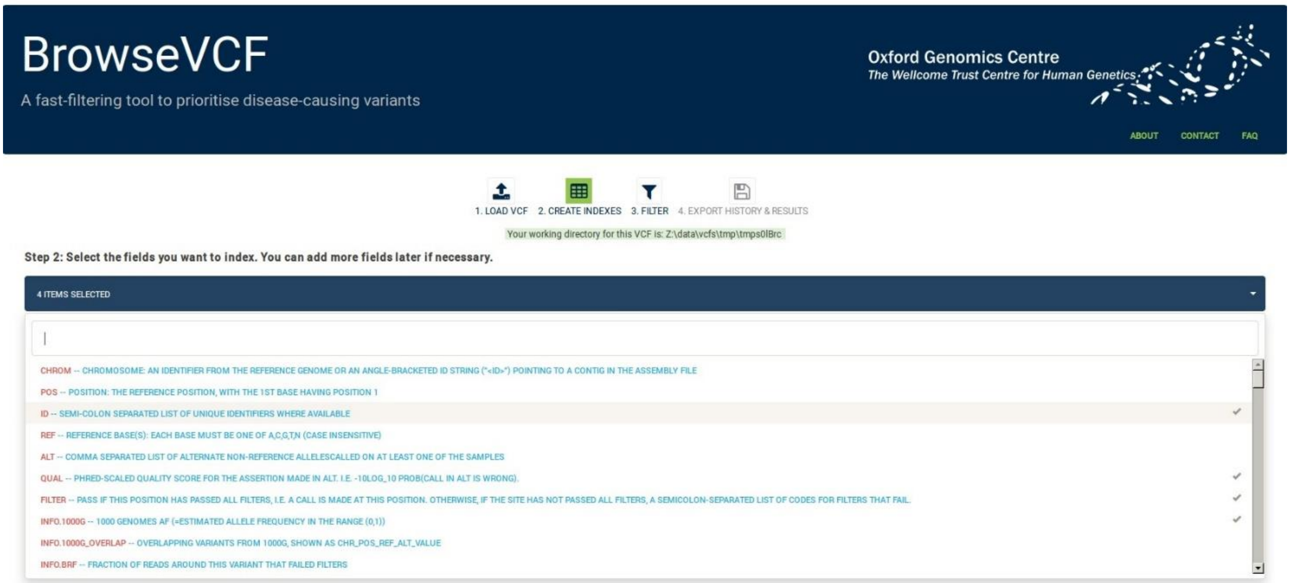
[14] http://www.gnu.org/licenses/gpl-3.0.en.html

[15] https://sites.stanford.edu/abms/giab

[16] Smedley D, Robinson PN. Phenotype-driven strategies for exome prioritization of human Mendelian disease genes. *Genome Med*. 2015 Jul 30;7(1):81.

Figures

Figure 1. Screenshot of BrowseVCF step 2 (index creation). The drop-down menu lets the user browse through the annotation fields present in the input VCF file and select one or more fields to be used in filtering. A text box allows keyword searching across fields.



BrowseVCF

A fast-filtering tool to prioritise disease-causing variants

Oxford Genomics Centre
The Wellcome Trust Centre for Human Genetics

ABOUT CONTACT FAQ

1. LOAD VCF 2. CREATE INDEXES 3. FILTER 4. EXPORT HISTORY & RESULTS

Your working directory for this VCF is: Z:\data\vcfs\tmp\tmps0lBrc

Step 2: Select the fields you want to index. You can add more fields later if necessary.

4 ITEMS SELECTED

CHROM -- CHROMOSOME: AN IDENTIFIER FROM THE REFERENCE GENOME OR AN ANGLE-BRACKETED ID STRING ("<ID>") POINTING TO A CONTIG IN THE ASSEMBLY FILE

POS -- POSITION: THE REFERENCE POSITION, WITH THE 1ST BASE HAVING POSITION 1

ID -- SEMI-COLON SEPARATED LIST OF UNIQUE IDENTIFIERS WHERE AVAILABLE ✓

REF -- REFERENCE BASE(S): EACH BASE MUST BE ONE OF A,C,G,T,N (CASE INSENSITIVE)

ALT -- COMMA SEPARATED LIST OF ALTERNATE NON-REFERENCE ALLELESCALLED ON AT LEAST ONE OF THE SAMPLES

QUAL -- PHRED-SCALED QUALITY SCORE FOR THE ASSERTION MADE IN ALT. I.E. -10LOG_10 PROB(CALL IN ALT IS WRONG). ✓

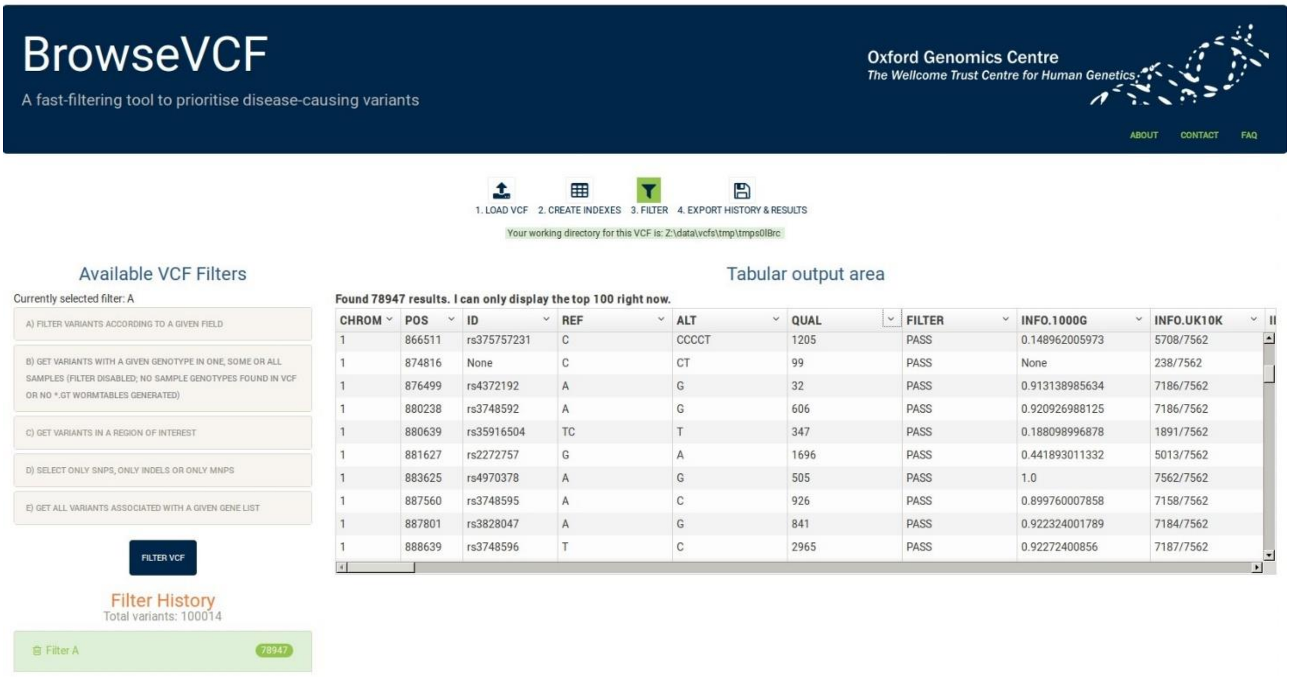
FILTER -- PASS IF THIS POSITION HAS PASSED ALL FILTERS, I.E. A CALL IS MADE AT THIS POSITION. OTHERWISE, IF THE SITE HAS NOT PASSED ALL FILTERS, A SEMICOLON-SEPARATED LIST OF CODES FOR FILTERS THAT FAIL. ✓

INFO.1000G -- 1000 GENOMES AF (=ESTIMATED ALLELE FREQUENCY IN THE RANGE (0,1)) ✓

INFO.1000G_OVERLAP -- OVERLAPPING VARIANTS FROM 1000G, SHOWN AS CHR_POS_REF_ALT_VALUE

INFO.BRF -- FRACTION OF READS AROUND THIS VARIANT THAT FAILED FILTERS

Figure 2. Screenshot of BrowseVCF step 3 (filtering). The left panel lists the five available filters, and expands to allow the user to define various filter options and cutoffs. The "Filter History" panel keeps track of all the sequential filters applied to the initial data and shows the output number of variants. The right panel is the output area, which displays the top 100 variants resulting from each consecutive filter. Fields (shown as columns) can be sorted and hidden if desired.



BrowseVCF

A fast-filtering tool to prioritise disease-causing variants

Oxford Genomics Centre
The Wellcome Trust Centre for Human Genetics

ABOUT CONTACT FAQ

1. LOAD VCF 2. CREATE INDEXES 3. FILTER 4. EXPORT HISTORY & RESULTS

Your working directory for this VCF is: Z:\data\vcfs\tmp\tmps0lBrc

Available VCF Filters

Currently selected filter: A

A) FILTER VARIANTS ACCORDING TO A GIVEN FIELD

B) GET VARIANTS WITH A GIVEN GENOTYPE IN ONE, SOME OR ALL SAMPLES (FILTER DISABLED; NO SAMPLE GENOTYPES FOUND IN VCF OR NO *.GT WORMTABLES GENERATED)

C) GET VARIANTS IN A REGION OF INTEREST

D) SELECT ONLY SNPS, ONLY INDELS OR ONLY MNPS

E) GET ALL VARIANTS ASSOCIATED WITH A GIVEN GENE LIST

FILTER VCF

Filter History
Total variants: 100014

Filter A 78947

Tabular output area

Found 78947 results. I can only display the top 100 right now.

CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO.1000G	INFO.UK10K
1	866511	rs375757231	C	CCCCT	1205	PASS	0.148962005973	5708/7562
1	874816	None	C	CT	99	PASS	None	238/7562
1	876499	rs4372192	A	G	32	PASS	0.913138985634	7186/7562
1	880238	rs3748592	A	G	606	PASS	0.920926988125	7186/7562
1	880639	rs35916504	TC	T	347	PASS	0.188098996878	1891/7562
1	881627	rs2272757	G	A	1696	PASS	0.441893011332	5013/7562
1	883625	rs4970378	A	G	505	PASS	1.0	7562/7562
1	887560	rs3748595	A	C	926	PASS	0.899760007858	7158/7562
1	887801	rs3828047	A	G	841	PASS	0.922324001789	7184/7562
1	888639	rs3748596	T	C	2965	PASS	0.92272400856	7187/7562