Bioinformatics, 36(19), 2020, 4951–4954
doi: 10.1093/bioinformatics/btaa628
Advance Access Publication Date: 5 August 2020
Applications Note

Sequence analysis

Seqminer2: an efficient tool to query and retrieve genotypes for statistical genetics analyses from biobank scale sequence dataset

Lina Yang[1], Shuang Jiang[2], Bibo Jiang[1], Dajiang J. Liu[1,*] and Xiaowei Zhan[2,*]

[1]Department of Public Health Sciences, Penn State College of Medicine, Hershey, PA 17033, USA and [2]Department of Clinical Science, University of Texas Southwestern Medical Center, Dallas, TX 75390, USA

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on March 3, 2020; revised on May 12, 2020; editorial decision on July 2, 2020; accepted on July 3, 2020

Abstract

Summary: Here, we present a highly efficient R-package *seqminer2* for querying and retrieving sequence variants from biobank scale datasets of millions of individuals and hundreds of millions of genetic variants. *Seqminer2* implements a novel variant-based index for querying VCF/BCF files. It improves the speed of query and retrieval by several magnitudes compared to the state-of-the-art tools based upon *tabix*. It also reimplements support for BGEN and PLINK format, which improves speed over alternative implementations. The improved efficiency and comprehensive support for popular file formats will facilitate method development, software prototyping and data analysis of biobank scale sequence datasets in R.

Availability and implementation: The *seqminer2* R package is available from https://github.com/zhanxw/seqminer. Scripts used for the benchmarks are available in https://github.com/yang-lina/seqminer/blob/master/seqminer2%20benchmark%20script.txt.

Contact: dajiang.liu@psu.edu or xiaowei.zhan@utsouthwestern.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1. Introduction

Human genetic studies are revolutionized by the cost-effective sequencing and genotyping technologies. As the cost for deep whole genome sequencing drops below $1000 USD and that for genome-wide genotyping drops below $100 USD, many ultra-large biobank scale datasets with millions of individuals begin to emerge. For sequence datasets with many individuals, the number of discovered variations is also increasing rapidly, as more rare variants can be uncovered as more samples are sequenced. For example, in the Trans-Omics Precision Medicine (TOPMed) sequencing project, there are >460 million variants identified from >100 000 whole genome deep sequenced individuals (Taliun *et al.*, 2019). In the UK biobank dataset, approximately half a million individuals were genotyped (Bycroft *et al.*, 2018). After genotype imputation using the Haplotype Reference Consortium panel, plus the UK10K and 1000 Genomes reference panel, there are over 80 million genetic variants. Given the scale of these sequence datasets, it is infeasible to directly read them into RAM. Efficient query and retrieval of information from these datasets have become a central problem that precedes virtually all genetic analyses.

An effective index is critical for efficient query and retrieval for sequence datasets. Conceptually similar to the index used in books and dictionaries, index for computer files allows us to narrow down the searches to the genomic intervals that may overlap the variants of interest. For most sequence data formats, such as VCF/BCF format, *tabix* (Li, 2011) index is the default choice for query which combines binning and linear index to query the files. *Tabix* works by clustering chromosomal positions into bins of fixed sizes. Therefore, the index file size is small and does not vary with the number of variants. For densely genotyped datasets with many individuals, each bin in the *tabix* index contains numerous variants. Even when querying and retrieving a single variant, the entire bin that contains the variant will be uncompressed and extracted, which greatly reduces the efficiency of retrieving variants. Existing R packages such as *VariantAnnotation* (Obenchain *et al.*, 2014) and *PopGenome* (Pfeifer *et al.*, 2014) integrate *tabix* index and can become very slow for biobank scale dataset.

There have been efforts to improve the query of large sequence datasets. One approach is to define more compact file formats as BGT (Li, 2016), GQT (Layer *et al.*, 2016), so that indexing and queries can be more efficiently performed. A major limitation for

these approaches is that they only support limited variant types. Multiallelic variants or imputed genotypes may not be supported well. Also these new file formats have not been adopted by many downstream analysis softwares, such as genetic association analysis. These limitations prevent them from being widely used in statistical genetics research.

Alternatively, to improve the query of sequence datasets, we seek to develop a novel variant-based index (*vbi*) that can work with VCF/BCF format, and provide implementations both in R and in command line tools for querying genomic sequence datasets. With the novel *vbi* index, *seqminer2* greatly improved existing R packages for querying VCF/BCF files. Compared to *VariantAnnotation* and *PopGenome*, the query speed for *seqminer2* can be magnitude faster. The companion *seqminer2* command line tool for querying VCF files outperformed *bcftools* (Li *et al.*, 2009) in speed by 5-folds when extracting single range from VCF file and by more than 200-folds when extracting multiple randomly chosen genomic ranges. It slightly outperformed *GIGGLE* (Layer *et al.*, 2018) in speed (*seqminer2* supported many other features). Such speed improvement makes it possible to retrieve variant genotypes on the fly for many applications such as the calculation of linkage disequilibrium (LD) coefficients, the calculation of LD scores, gene-level association analysis and transcriptomic wide association analysis. In addition to improved speed, *seqminer2* also requires much smaller memory to run compared to other tools when querying or retrieving sequence variants.

Another uniqueness of *seqminer2* is its comprehensive support for popular file formats. To make *seqminer2* a convenient and comprehensive tool, we also reimplemented support for a few other file formats that are commonly used in statistical genetics analysis, including BGEN (Band and Marchini, 2018) format, which was developed to store imputed genotypes for large datasets and binary PLINK format, which is a state-of-art file format for storing array genotypes. Our implementation was considerably faster than the *rbgen* package for querying BGEN files, and of comparable speed as *BEDMatrix* for querying binary PLINK files. To our knowledge, *seqminer2* is the only package that supports all commonly used file formats in statistical genetics analysis (Table 1). Its comprehensive feature and improved efficiency make it a valuable tool for data analysis and method development in R.

2 Materials and methods

2.1 Motivation for the development of *VBI*

Tabix is widely used tool to randomly query and retrieve sequence variants from large-scale genomic datasets. It uses a hybrid of binning and linear index to query bgzip compressed VCF/BCF or generic tab-delimited files. Binning index is small in size and efficient for querying moderate-sized VCF files with a few thousand individuals and millions of genetic variants. Yet, for densely genotyped biobank scale datasets, *tabix* index become very inefficient owing to the design of binning index.

Table 1. Feature comparison for *seqminer2, VariantAnnotation, PopGenome, giggle, bcftools, rbgen, snpStat* and *BEDMatrix*

Software	Query file types	Integration with R
seqminer2	VCF, BCF, BGEN, PLINK	Yes
VariantAnnotation	VCF	Yes
PopGenome	VCF	Yes
Giggle	VCF, PLINK	No
Bcftools	VCF, BCF	No
Rbgen	BGEN	Yes
snpStat	PLINK	Yes
BEDMatrix	PLINK	Yes
PLINK2[a]	PLINK	No

[a]PLINK2 does not directly support VCF/BCF files. Instead, PLINK2 supports VCF/BCF by converting them first to binary BED files.

For binning index, the genetic variants in each chromosome are clustered hierarchically into bins of different sizes. When querying a genetic interval, the smallest bin that contains the queried interval will be uncompressed and processed. The bin size is preset and not dependent on the density of the variants across the chromosome. For modern sequence datasets with many individuals, there can be one genetic mutation observed per 20 basepair (Taliun *et al.*, 2019). In this case, for a sequence dataset with 100 000 individuals, the smallest bin of 16 384 bp used by *tabix* contains 80 MB data. So to query 100 randomly chosen genetic variants, 100 bins with 80 000 genetic variants and 8 GB data will need to be uncompressed, even though most of the uncompressed data is irrelevant to the queried variants.

The limitation for binning index motivated us to develop an alternative method to query VCF/BCF index.

2.2 Build an index for bgzipped VCF/BCF file

As *tabix* index, *vbi* is also based upon bgzip file. In *vbi* index file, we store the genomic position and the offset for each variant in the index file. So for a file with M variants, all the indices form a M by two matrix. For a chromosome with 10 million markers, the index file size is 107*2* (8 byte) = 160 MB. The size of the *vbi* index file does not depend on the number of samples, and remains small enough to be loaded to computer RAM in entirety. With *vbi*, we can directly locate the location of the queried genetic variants, and minimize the amount of redundant data that needs to be uncompressed.

2.3 Query bgzip compressed VCF/BCF by positions

To locate any marker in a large VCF/BCF file by position, we make use of binary search, which takes no more than log2(number of marker) comparisons. It means that for a dataset with 10 million genetic variants, it takes only up to 24 comparisons. In this case, to query 100 random selected genomic ranges from a VCF file, this index strategy takes less than 20 s, while the alternative *tabix* index takes 4 min, 12 times slower. To query 100 randomly chosen non-consecutive ranges from a BCF file, this index strategy takes less than 1 s.

2.4 Reimplementation of BGEN support

BGEN format was developed to store genotype probabilities from genotype imputation. It has become a popular file format in statistical genetic analysis. For example, imputed genotypes from UK Biobank were released in BGEN format. We followed the BGEN file format specification and reimplemented the query and retrieval in C++. This reimplementation is in fact considerably faster than the official implementation. We attribute the speed improvement to several software engineering advances:

First, we seek to minimize disk I/O, which is a major bottleneck for query and retrieval speed. For example, to read multiple genomic ranges, we first cluster these ranges, and merge ranges in proximity and read the whole block of the variants of all samples into the memory. Second, our implementation improves by optimizing the most common cases. BGEN format, as developed is very general, and capable of handling multiploidy and multiallelic variants. Yet, human germline genome is diploid, and a majority of the variants are biallelic. We improve the performance by simplifying the implementation on the most common case, and then separately considering the more special cases. Compared to the original implementation that treats common and uncommon cases indifferently, *seqminer2* gains considerable efficiency. Lastly, we used the dictionary-based decompression algorithm in ZSTD, which has faster decompression speed compared to the system-default decompression algorithm for large files. Together these software engineering works considerably improved the speed of *seqminer2* over *rbgen* package.

3. Results

We extensively evaluated *seqminer2* for querying VCF/BCF, BGEN and PLINK files using both the R package and the command line tool. We considered scenarios with very large number of individuals ($N = 487\,409$) as motivated by UK Biobank datasets (Bycroft *et al.*, 2018). We compared the query of various numbers of genomic ranges from largest chromosome (chr2 with number of markers $M = 8\,129\,063$) and smallest chromosome (chr21 with number of markers $M = 1\,261\,158$). Comparisons were conducted on a computer server with Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80 GHz CPU, 128 GB RAM and 7200 rpm hard drive. We also conducted comparisons requesting only 4 GB of memory in each node.

The time used for querying and parsing the output was recorded. For each scenario, the query was repeated 100 times to ensure stability of the results. The median time used was reported.

3.1 Evaluation of reading VCFs with variant-based index

First, we evaluated *seqminer2* against *VariantAnnotation* and *PopGenome* R packages to query *tabix*-indexed VCF files. For a file containing 487 409 individuals and 8 129 063 genetic variants, the function *seqminer::readSingleChromosomeVCFToMatrixByRange* took 22.35 s to read a single range with 100 genetic variants, and 16.57 s to read 100 randomly chosen ranges. The time used for querying multiple randomly selected ranges is often comparable and sometimes even less than the time used for querying one single range, which represents a unique advantage for *vbi* index.

On the other hand, the R packages based upon *tabix* index have greatly reduced speed for querying multiple randomly selected ranges. This is because randomly chosen ranges tend to fall into multiple distinct bins, each of which contains a large amount of data. Retrieval of variants in these ranges requires uncompressing all overlapping bins, which constitutes a severe bottle neck.

In our comparison, we used *readGT* function in *VariantAnnotation* package Version 1.28.11 to query variants. It took 99.86 s to read a single range with 100 genetic-variants and 355.40 s to read 100 randomly selected ranges from the same file as we used for *seqminer2*. We also compared with *readVCF* function in *PopGenome* package. It took 830.70 s to read a single range with 100 genetic variants. To the best of our knowledge, *readVCF* does not support the query of multiple *tabix* ranges. If simply looped through all 100 randomly chosen ranges, the query took more than 21 h. The advantage of *seqminer2* over *VariantAnnotation* and *PopGenome* increased as more ranges were queried and retrieved (Table 2). For instance, *seqminer2* took 187.62 s to read a single range with 1000 genetic variants, whereas *VariantAnnotation* took 1006.10 s which was more than 5 times of *seqminer2*, and *PopGenome* took 802.90 s. When reading 1000 randomly chosen ranges, *seqminer2* took 166.60 s, but *VariantAnnotation* and *PopGenome* respectively took 4035.86 s and >1 day, which was substantially slower than *seqminer2*.

We next evaluated *seqminer2* command line tool against two other command line tools, *GIGGLE* and *bcftools*. *Seqminer2* outperformed *GIGGLE* when reading a single range or randomly selected ranges. *Bcftools* was far less efficient than *seqminer2*. It took up to 4 times more time when reading a single range, and 200 times more time than *seqminer2* when reading randomly selected ranges (Supplementary Table S1A).

3.2 Evaluation of reading BCF files

Seqminer2 is one of the few tools that support BCF format, which is a binary version of the VCF format. BCF format was developed as an improvement to VCF files for more efficient storage and query. Both *seqminer2* and *bcftools* performed efficiently while reading a single region. They took less than 1 s to extract 100 variants and less than 5 s to extract 1000 variants, which was indeed much faster compared to reading VCF files. However, when reading multiple randomly selected ranges, there was a sharp increase in time for *bcftools*. *Bcftools* took up to 23 min to extract 1000 randomly chosen ranges, a disadvantage owing to the *tabix* index, while *seqminer2* still only took less than 5 s (Table 3).

3.3 Evaluation of reading binary PLINK files

We used the function *readPlinkToMatrixByIndex* in *seqminer2* to query binary PLINK files. *Seqminer2* and *BEDMatrix* (Grueneberg, 2019) performed about equally well when read in a single range or multiple randomly selected ranges. The function *read.plink* in *snpStats* (Clayton, 2019) failed to read in variants, which revealed its limitations for handling biobank scale datasets.

Unlike *seqminer2*, *PLINK2* can only output extracted sequence variants from BED files into a text file, which needs to be read into R separately. This appears to be less convenient. In our evaluation, *seqminer2* is also faster than *PLINK2* in almost all scenarios for querying and reading variants into R (Supplementary Table S1B).

3.4 Evaluation of reading BGEN files

We used *seqminer2* and an R package specifically designed to load BGEN format files called *rbgen* to query bgenix-indexed bgen files. The function *readBGENToMatrixByRange* in *seqminer2* was >10 times faster than *bgen.load* in *rbgen* to read a single range. When reading randomly chosen ranges, the advantage for *seqminer2* was even bigger: *seqminer2* was >200 times faster than *rbgen* when reading randomly chosen ranges from largest chromosome in UK Biobank, and remained >10 times faster when reading from smallest chromosome in UK Biobank (Supplementary Table S1C).

3.5 Evaluation of memory usage

We recorded the maximum memory each software used when querying files with the largest chromosome (chromosome 2) in the UK Biobank dataset. *Seqminer2* almost always required less memory than other tools (Supplementary Table S2).

We also benchmarked the performance of each software using computers with small memories. Specifically, we requested only 4 GB of memory in each node in the cluster when testing each software. In this setting, *seqminer2* still remained the fastest tool. When extracting variants from VCF files, *VariantAnnotation* and *PopGenome* failed to run, because they required much more RAM than 4 GB to retrieve 1000 variants. *Seqminer2* was the only R package that extracted variants successfully (Supplementary Table S3).

Table 2. Comparison of query speed of *seqminer2* with VCF files

Datasets	Total number of queried variants	Single range			Multiple random genomic ranges		
		seqminer2	VariantAnnotation	PopGenome	seqminer2	VariantAnnotation	PopGenome
chr2	100	22.35	99.86	830.70	16.57	355.40	78253.79
	1000	187.62	1006.10	802.90	166.60	4035.86	>1d
chr21	100	20.48	98.22	860.15	21.15	395.70	81948.25
	1000	193.92	979.51	903.74	250.93	5401.41	>1d

Note: Sample size is $N = 487\,409$. We considered scenarios (i) for querying a single genomic range that contains 100 or 1000 variants and (ii) for querying 100 or 1000 randomly chosen genomic ranges with 1 variant in each range. For each scenario, we repeat the comparison for 100 times and median time in second was recorded.

Table 3. Comparison of query speed of *seqminer2* with BCF files

Datasets	Total number of queried variants	Single range		Multiple random genomic ranges	
		seqminer2	bcftools	seqminer2	bcftools
chr2	100	0.75	0.51	0.99	128.35
	1000	2.73	4.65	4.30	1396.00
chr21	100	0.25	0.18	0.63	125.04
	1000	1.21	2.02	2.80	1296.98

Note: Sample size is $N = 487\,409$. We considered scenarios (i) for querying a single genomic range that contains 100 or 1000 variants and (ii) for querying 100 or 1000 randomly chosen genomic ranges with 1 variant in each range. For each scenario, we repeat the comparison for 100 times and median time in second was recorded.

4. Conclusions and discussions

In this article, we showed that *seqminer2* is a very efficient software tool optimized for indexing and querying VCF/BCF. It also accommodates other commonly used file format in statistical genetic analysis, such as BGEN and PLINK files. It can scale well to biobank scale sequence datasets with hundreds of millions of variants.

While *seqminer2* greatly improves the efficiency over other tools, a few practical considerations in data analysis can also make huge differences in speed and warrants discussions. Disk I/O is a major bottleneck for large scale data analysis. Minimizing disk I/O is key to improving the speed for data analysis. If multiple ranges of data (e.g. multiple genes) need to be analyzed, we recommend reading in multiple ranges in one batch using *seqminer2*, instead of reading each range separately. If the goal is to analyze the entire chromosome, as long as the system memory allows, reading in all variants on the chromosome can be more efficient.

The current implementation focuses widely used file formats VCF/BCF, BGEN and PLINK. We noted that the indexing strategy can be readily extended to support generic file formats, e.g. tab-delimited files that include columns of chromosomal positions. Generic file formats have been broadly used in store annotation information, GWAS summary statistics. Implementing *vbi* index could be extremely useful to accelerate the query of files of these files.

In conclusion, the improved efficiency and comprehensive features of seqminer2 will greatly facilitate method development and analysis in R. We expect it to be a very useful tool for biobank scale data analysis.

Funding

The work is supported by grants R01HG008983 from NIH/NHGRI and R01GM126479 from NIH/NIGMS.

Financial Support: none declared.

Conflict of Interest: none declared.

References

Band,G. and Marchini,J. (2018) BGEN: a binary file format for imputed genotype and haplotype data. (preprint) https://www.biorxiv.org/content/10.1101/308296v2.

Bycroft,C. *et al.* (2018) The UK Biobank resource with deep phenotyping and genomic data. *Nature*, **562**, 203–209.

Clayton,D. (2019) snpStats: SnpMatrix and XSnpMatrix classes and methods. https://www.bioconductor.org/packages/release/bioc/html/snpStats.html

Grueneberg,A. (2019) BEDMatrix: extract genotypes from a PLINK.bed file. https://rdrr.io/cran/BEDMatrix/man/BEDMatrix-package.html

Layer,R.M. *et al.* (2016) Efficient genotype compression and analysis of large genetic-variation data sets. *Nat. Methods*, **13**, 63–65.

Layer,R.M. *et al.* (2018) GIGGLE: a search engine for large-scale integrated genome analysis. *Nat. Methods*, **15**, 123–126.

Li,H. (2011) Tabix: fast retrieval of sequence features from generic TAB-delimited files. *Bioinformatics*, **27**, 718–719.

Li,H. (2016) BGT: efficient and flexible genotype query across many samples. *Bioinformatics*, **32**, 590–592.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Obenchain,V. *et al.* (2014) VariantAnnotation: a Bioconductor package for exploration and annotation of genetic variants. *Bioinformatics*, **30**, 2076–2078.

Pfeifer,B. *et al.* (2014) PopGenome: an efficient Swiss army knife for population genomic analyses in R. *Mol. Biol. Evol.*, **31**, 1929–1936.

Taliun,D. *et al.* (2019) Sequencing of 53 831 diverse genomes from the NHLBI TOPMed Program. *bioRxiv. doi:10.1101/563866v1*.