fastman: A fast algorithm for visualizing GWAS results using Manhattan and Q-Q plots

Soumya Subhra Paria[a], **Sarthok Rasique Rahman**[a], **Kaustubh Adhikari**[a]
[a]The Open University, UK
[b]The University of Alabama, Tuscaloosa, Alabama, USA

Abstract. Visualization of GWAS summary statistics, specifically P-values, as Manhattan plots is widespread in GWAS publications, and many popular software tools are available, such as the R package *qqman*. But there is substantial need for further development, such as the handling of non-human data. We provide a new R package, *fastman*, with major additional capabilities. It handles genomes of non-model organisms, even those at a draft stage, i.e. contigs that haven't been compiled to chromosomes. Non-numeric chromosome IDs are supported. It supports plotting of other genetic scores, such as FST, D statistics, selection statistics such as PBS, or other kinds of GWAS statistics such as beta. Importantly, negative or two-tailed values are supported in this package. We implement a heuristic algorithm that drastically reduces plotting time for huge datasets without any loss of visual precision, while allowing for many different data types and missing data. We provide substantial additional flexibility in highlighting and annotation. In summary, we have developed a package *fastman* in R for fast and efficient visualization of GWAS results and other genomewide scores using Manhattan and Q-Q plots. The package can create plots directly from association outputs by *PLINK*. Alternatively, it can produce plots from any R data frame with custom columns and is equipped to handle big datasets with fast plot generation. It is available for public use on https://github.com/kaustubhad/fastman.

1 Introduction

In recent years, the field of human genetics has grown immensely in terms of data production and analysis.[9,10] A typical chip used to genotype one participant contains half a million to one million genetic variants. Subsequently, imputation – a statistical method to probabilistically infer genotype data of additional variants based on the chip genotype information – increases the number of usable variants to up to 10 million.[10] Proportional to the rise in the use of big datasets, there has been a rise in the development of fast algorithms for their handling, processing and visualization, for big datasets and large output files produced by the analyses of them.[11–14] An example of this is the widespread use of software for visualization of GWAS summary statistics[2,15]

The objective of GWAS is to understand the association of genotypes with phenotypes. This is performed by testing for differences in the allele frequency of genetic markers between individuals with similar ancestries but different phenotypes. In GWAS, hundreds of thousands of genetic markers across many genomes are tested to find which of the variants has statistically significant association with a particular trait or disease of interest. The most popularly studied genetic variants in GWAS are single-nucleotide polymorphisms (SNPs), which are germline substitutions of single

nucleotides at specific positions in the genome. The testing is done by performing multiple linear regressions on SNPs as well as basic covariates like age, sex etc. A standard set of GWAS results consists of a list of SNPs, their associated chromosomal position, and a P-value which represents the statistical significance of the association of interest.

One of the most popular ways of visualizing GWAS results is the Manhattan plot. This is essentially a plot of the negative logarithms of P-values on the y-axis versus the chromosomal position of the SNP on the x-axis. Hence each dot in a Manhattan plot represents a SNP. Stronger the association, smaller is the P-value, and higher is the value of the negative logarithm. Hence SNPs with the highest associations are positioned at the top of the graph, giving the plot the appearance of a Manhattan skyline, a group of skyscrapers rising above the standard buildings. Another popular plot for viewing GWAS results is the quantile-quantile plot or the Q-Q plot. In the Q-Q plot, the observed P-values for each SNP are sorted from largest to smallest and plotted against the expected uniform distribution of P-values under the null hypothesis (zero association). If the observed values correspond to the expected values, all points are on or near the diagonal line. If a SNP has a statstically significant association, then the observed P-values corresponding to that SNP will move towards the y-axis.

There are several softwares and R packages available for visualizing GWAS results using Q-Q and Manhattan plots. These plots can be created using standalone desktop software *Haploview*,[5] or for focused regions using the web-based application *LocusZoom*.[6] The most popular R package for generating the Q-Q and Manhattan plots directly from *PLINK*[1] result files is *qqman*.[2] *qqman* is an R package that contains two functions, *manhattan()* and *qq()* which can be used to generate the respective plots. Both the functions take a data frame with columns containing the chromosome number, chromosomal position and P-value as input. The typical input should be in the form of a *PLINK*–assoc output, but the package offers users the flexibility of using a custom dataframe while specifying the required column names. The default output is a black and white plot with horizontal lines drawn at $-\log_{10}(1 \times 10^{-5})$ for "suggestive" associations and $-\log_{10}(5 \times 10^{-8})$ for the "genome-wide significant" threshold. The user has the option to change the colour scheme and to change the threshold levels. The function also gives the option to highlight SNPs of choice, if the user can provide a column of SNPs to highlight. The package also offers a wide range of annotation options for the user, and it being available as an R package, the user can use R to control

every granular aspect to customize the output to their choice.

Despite the plethora of options provided by *qqman*, it lacks versatility in terms of input handling. *qqman* only accepts P-values as inputs, and it is not compatible with other genome-wide population genetic parameters like FST, pi and D statistics. These scores might represent a wide variety of genetic parameters like degree of association or measure of genetic differences and are very important for several genetic studies. Since these are scores, the ranges and distribution of their values are quite different from that of P-values. As *qqman* doesn't support non-numeric entries for chromosome number, results from genomes of non-model organisms cannot be plotted using this. Also, *qqman* cannot handle missing values in the input dataframe. Another drawback of *qqman* is the time for plot generation. On a typical imputed *PLINK* assoc file of 10 million SNPs, *qqman* takes 737 seconds for generating a Manhattan plot. Therefore, we have developed a package *fastman* in R for fast and efficient visualization of GWAS results using Q-Q and Manhattan plots. This package creates the plot directly from assoc outputs provided by *PLINK*, which is one of the most popular software used for GWAS. In addition to a standard *PLINK* output, the package can produce the plots from any data frame with chromosome, position and P-value, and is equipped to handle big data with fast plot generation and optimized memory usage.

2 Implementation

fastman is an R package for fast and efficient visualizing of GWAS results using Q-Q and Manhattan plots directly from *PLINK* output files. The package contains two functions *fastman()* and *fastqq()*, the first one for visualizing Manhattan plots and the other for Q-Q plots.

2.1 Features and Functionality

The main features of the package are:

- **Speed**: It drastically reduces time in plot generation compared to *qqman*, which is the most popular existing package used for this purpose. On a typical imputed *PLINK* assoc file of 10 million SNPs, plotting time is reduced from 737s in *qqman* to 60s. We expect similar reduction in other datasets as well. We have provided a detailed speed comparison of *fastman* with other packages in the Results section.

- **Efficiency**: Memory management has been optimized to reduce space usage during plotting. A detailed comparison has been provided in the Results section.

- **Versatility**: It can handle various inputs ranging from P-values, logarithms of P-values to FST scores. It is compatible plotting with other genome-wide population genetic parameters (e.g., FST, pi and D statistics). It also has provision to allow both-sided scores, e.g., scores with negative values.

- **Non-model friendliness**: Apart from the GWAS results, our package additionally supports plotting of results from genomes of non-model organisms (often with hundreds of contigs or many scaffolds) and provides alphabetical and other ordering options.

- **Annotation and highlight versatility**: It provides the user a wide set of options to customize annotating and highlighting SNPs of interest.

- **Familiarity**: We understand that *qqman* is a very popular package for visualizing GWAS results. Keeping that in mind, we have kept a very similar set of input arguments and code structure compared to *qqman* to maintain a high degree of familiarity for the user.

2.2 Example Usage

The *fastman* package includes functions for creating Manhattan plots and Q-Q plots from GWAS results. For the first illustration (Figure 1) of GWAS data visualization, we use data from a GWAS analysis conducted on human participants, which is probably the most common kind of GWAS studies. We use a set of published GWAS summary statistics (link), which were released with the publication of a GWAS study[3] conducted on 6000 participants from various countries of Latin America. The participants were genotyped on a commercial Illumina OmniExpress genotyping chip containing 700K genomic variants. The data was subsequently imputed, based on the publicly available reference database called 1000 Genomes, to increase the number of available genetic variants to 10 million, using the software *Impute2*.[7] The particular association results used here correspond to the phenotype of facial hair density, which was used here since it contains a clear, prominent association peak on chromosome 2, as well as other suggestive and genome-wide significant association peaks. So this allows the display of both the suggestive and genome-wide

significant association threshold lines, as well as the Y-axis clipping performed by the maxP parameter.

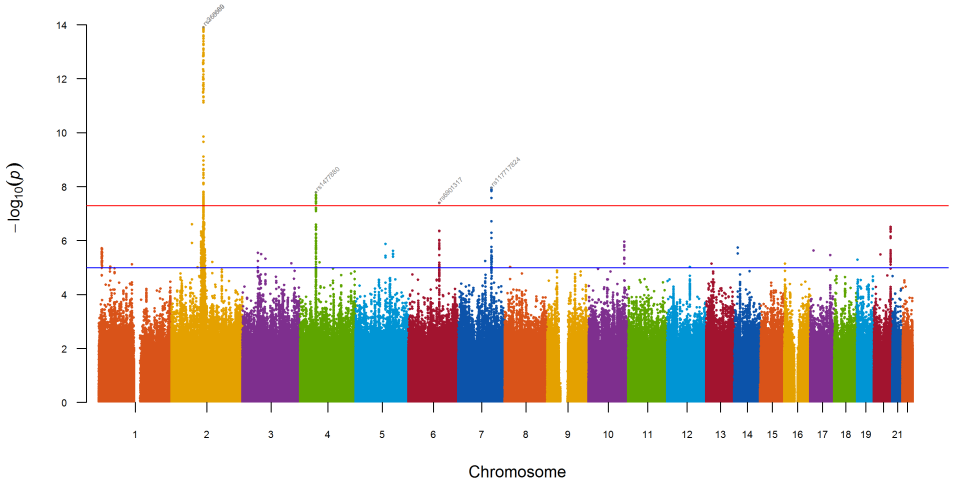

Fig 1 A sample Manhattan P-value plot from *fastman()*

The package allows the user to choose different annotation colours for different SNPs. The user has the option to colour only the part of the plot above the P-value threshold, while the rest of the plot stays grey. The following example (Figure 2) shows such a plot using the previous data. Here we have chosen to annotate the top SNP for every 20 megabase window.

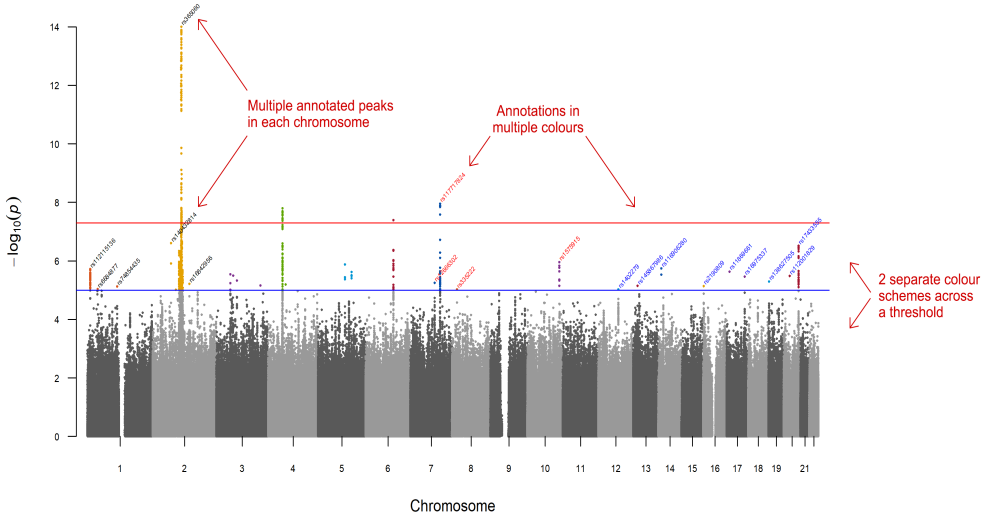

Fig 2 A detailed annotated Manhattan P-value plot from *fastman()*

Most of the commonly used GWAS visualization software like *qqman* are not able to handle data from non-model organisms. The genetic sequencing of non-model organisms is becoming rapidly popular, as new technologies enable low-cost high-quality sequencing these days. A particular challenge of working with non-model organisms is that a standardized reference genome is usually not available. To do the genetic sequencing, the genome is divided into small fragments of manageable size, and the fragments are then sequenced. These fragments need to be assembled properly so that the genomic pattern in a region can be established. However, due the lack of a standardized reference genome, such assembly is usually incomplete, or at a draft stage, which means that the data cannot be neatly organized into chromosomes, as is possible for humans and other model organisms. Therefore, the the small fragments or contigs are directly used in place of chromosomes for analysis and visualization, so that association results are referred to specific variants and specific contigs. Often a draft assembly will contain many thousands of contigs with custom names containing both alphabets and numerals as well as symbols such as. Such contig names also cause problems for many software.

For the third visualization example (Figure 3), we use a SNP data set from a published literature that utilized a bumble bee (Bombus terrestris) genome assembly (RefSeq GCA_000214255.1; consists of 10,672 contigs with contig N50 of 76,043 bp) to investigate the genetic basis of a lab-generated (yellow) color mutant.[4] The extent of genetic differentiation between two color morphs (wildtype black vs. mutant yellow) was assessed using the FST statistic, which is a score measuring the amount of differentiation. We plot the values of this FST statistic across the genome as a demonstration of the *fastman* package's capability of plotting score statistics, moving beyond just P-values.

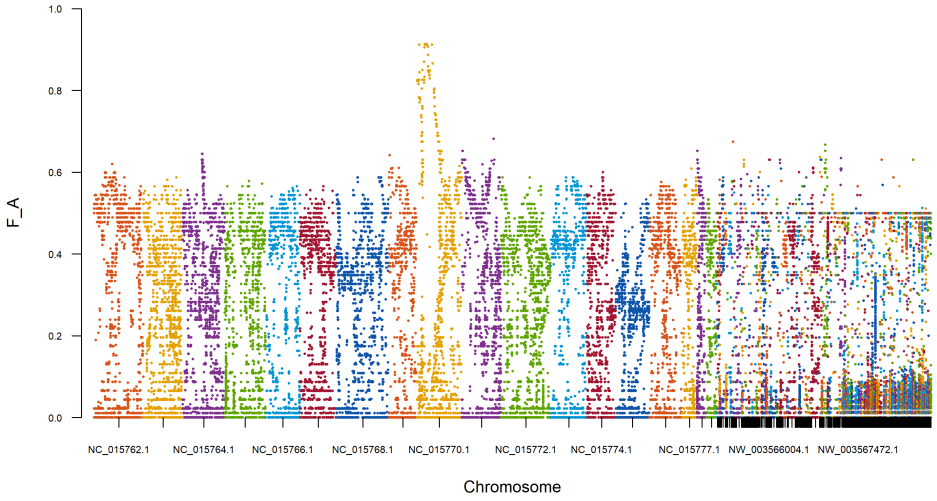

Fig 3 A Manhattan Fst score plot from *fastman()*

The *fastqq()* function accepts a vector of P-values as its input and provides a Q-Q plot with genomic inflation factor. For this illustration (Figure 4) we use P-values of the data set from our first example.

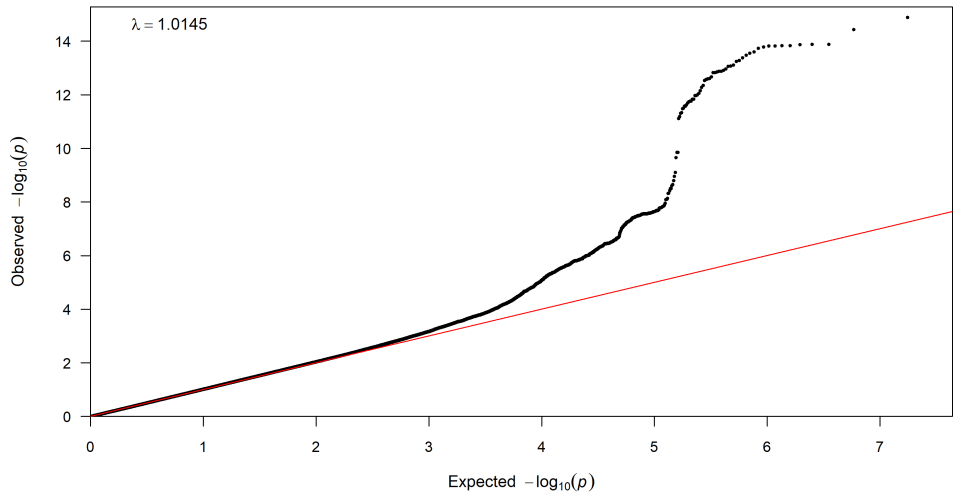

Fig 4 A sample Q-Q plot from *fastqq()*

3 Methods

The *fastman* package drastically reduces time in plot generation compared to *qqman*. We achieve this through a speedup algorithm where we reduce the size of the data for fast plotting. We assume

that the user is more interested in the tail (top/bottom/both) 0.2% of the data, and thus they are rounded to 3 digits, while the rest of the data is rounded to 2 digits. This shrinks the data size drastically, thereby reducing the plotting time. By visual inspection, we decided that we can round up to 3 digits in a high resolution image without observing any visible change in the less dense parts of the image, and the threshold is 2 digits for denser parts of the image. As the tail part of the distribution is plotted at the top of the bars in Manhattan plot, we are using a rounding off threshold of 3 digits for the tail and 2 digits for the rest.

If the data has less than 100k rows, then the full data is rounded to 3 digits, as the rest of the speedup procedure will not be significant in such a small data set. Else, we identify which of the tails of the data are significant, and we round that off to 3 digits. We understand that the data might have a significant right or left tail or both depending on the distribution of the score being plotted. For example, we expect the negative logarithm of P-values to follow exp(1) distribution with a significant right tail. Hence, in case of P-values, only the right tail is rounded to 3 digits while the rest of the data is rounded to 2 digits. We use the following statistics to measure the significance of the tails.

$R = \frac{P_{100} - P_{99.8}}{P_{99.8} - P_{50}}$ is used to measure the significance of the right tail.

$L = \frac{P_0 - P_{0.2}}{P_{0.2} - P_{50}}$ is used to measure the significance of the left tail.

P_i here refers to the i^{th} percentile. If the data has a significant tail, then we expect the corresponding statistics to take a value higher than 0.1. The cut-off value has been obtained heuristically by looking at distribution of various scores and P-values.

The user has the option to cancel the speedup procedure by specifying the same in the input parameters.

4 Results

We compared the performance of *fastman* with the existing packages in terms of speed and memory usage. The results are provided below.

4.1 Time

We use a dataset of 1,222,628 rows. We run the codes in a local system with the following specifications (1.60 GHz and 1.80 GHz processor, 15.8 GB usable RAM). We ran ten iterations on this

dataset. The median time taken by the packages across the iterations have been reported in Figure 4 below.

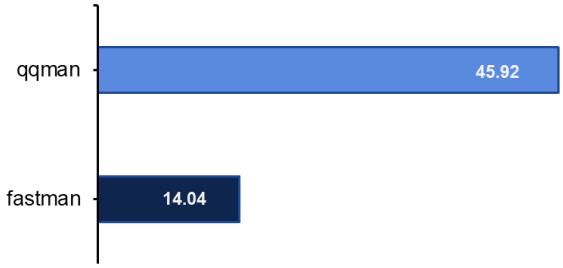

Fig 5 Time comparison of *fastman* and *qqman*

We observe that *fastman* reduces the time of plot generation drastically compared to *qqman* in this data. We expect the package to show similar behaviour in other datasets as well.

4.2 Memory

Using the same dataset, we compared the memory usage of the two packages. As demonstrated in Figure 5 below, *fastman* seems to take marginally higher memory space than *qqman* in spite of the additional calculations used for speeding up.

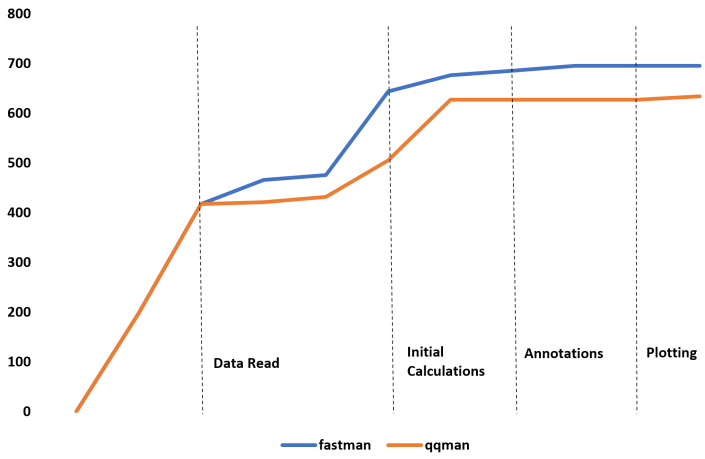

Fig 6 Memory usage comparison of *fastman* and *qqman*

Each dotted line represents the end of a specific stage of the code. The memory usage increases as the code reads the data and prepares the data for plot generation, after which we reach a plateau.

5 Conclusions

fastman is an R package that offers the user a fast and efficient option for visualizing GWAS results using Q-Q and Manhattan plots. While this can also be accomplished using various other software packages, *fastman* provides the user a plethora of input flexibility and output customization options while reducing the plot generation time drastically. The next significant future step in the *fastman* package will be creating a new function to implement Miami plots. A Miami plot is a plot where two different GWAS results are plotted along the same X axis, with one being plotted along the positive y-axis and the other along the negative side. It is called a Miami plot as it resembles the Miami skyline with the reflection of skyscrapers on the sea.

References

1 Shaun Purcell, Benjamin Neale, and Pak C. Sham, PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses, *The American Journal of Human Genetics*

2 Stephen D. Turner, qqman: an R package for visualizing GWAS results using QQ and manhattan plots, *Journal of Open Source Software*

3 Kaustubh Adhikari, Tania Fontanil, and Santiago Cal, A genome-wide association scan in admixed Latin Americans identifies loci influencing facial and scalp hair features, *Nature Communications*

4 Sarthok Rasique Rahman, Jonathan Cnaani, Lisa N. Kinch, Nick V. Grishin, and Heather M. Hines, A combined RAD-Seq and WGS approach reveals the genomic basis of yellow color variation in bumble bee Bombus terrestris, *Scientific reports*

5 J. C. Barrett, B. Fry, and M. J. Daly, Haploview: analysis and visualization of LD and haplotype maps, *Bioinformatics*

6 Randall J. Pruim, Ryan P. Welch, and Cristen J. Willer, LocusZoom: regional visualization of genome-wide association scan results, *Bioinformatics*

7 Bryan Howie, Christian Fuchsberger, Matthew Stephens, Jonathan Marchini, and Gonçalo R. Abecasis, Fast and accurate genotype imputation in genome-wide association studies through pre-phasing, *Nature Genetics*

8 Bryan Howie, Christian Fuchsberger, Matthew Stephens, Jonathan Marchini, and Gonçalo R. Abecasis, Fast and accurate genotype imputation in genome-wide association studies through pre-phasing, *Nature Genetics*

9 Emil Uffelmann, Qin Qin Huang, and Danielle Posthuma, Genome-wide association studies, *Nature Reviews Methods Primers*

10 Peter M. Visscher, Naomi R. Wray, and Qian Zhang, 10 Years of GWAS Discovery: Biology, Function, and Translation, *The American Journal of Human Genetics*

11 Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee, Second-generation PLINK: rising to the challenge of larger and richer datasets, *GigaScience*

12 Bongsong Kim, Xinbin Dai, Wenchao Zhang, Zhaohong Zhuang, Darlene L Sanchez, Thomas Lübberstedt, Yun Kang, Michael K Udvardi, William D Beavis, Shizhong Xu, and Patrick X Zhao, GWASpro: a high-performance genome-wide association analysis server, *Bioinformatics*

13 Greg R. Ziegler, Ryan H. Hartsock, and Ivan Baxter, Zbrowse: an interactive GWAS results browser, *PeerJ Computer Science*

14 Marvin N. Wright and Andreas Ziegler, ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R, *Journal of Statistical Software*

15 H Daniel, fastman, https://github.com/danielldhwang/fastman

16 Soumya Subhra Paria and Kaustubh Adhikari, fastman, https://github.com/kaustubhad/fastman