GOAT : Genetic Output Analysis Tool

An open source GWAS and genomic region visualisation tool.

Beatriz Kanzki, Victor Dupuy, Cedric Urvoy, Fodil Belghait, Alain April
École de Technologie Supérieure (ÉTS)
1100, rue Notre-Dame ouest,
Montréal, QC, Canada
beatriz.kanzki.1@ens.etsmtl.ca
victor.dupuy.1@ens.etsmtl.ca
cedric.urvoy.1@ens.etsmtl.ca
fbelghait@gmail.com
alain.april@etsmtl.ca

François Harvey, François- Christophe Marois-Blanchet, Michael S. Phillips, Johanne Tremblay and Pavel Hamet
Centre de Recherche du Centre Hospitalier de l'Université de Montréal, CRCHUM
900, rue Saint-Denis,
Montréal, QC, Canada
francois.harvey.chum@ssss.gouv.qc.ca
françois-christophe.marois-blanchet.chum@ssss.gouv.qc.ca
pgxdoc@gmail.com
johanne.tremblay@umontreal.ca
pavel.hamet@umontreal.ca

***Abstract*— Genome wide association studies (GWAS) are a widely used approach in genetic research to identify genes or genetic variants involved in human diseases. Each GWAS examines millions of unique single nucleotide polymorphisms (SNPs) at the same time that are associated with phenotypic traits and diseases. In the context of identifying complex associations in large patient cohorts, this type of study involves a vast amount of clinical and genetic data. In order to analyze these complex datasets efficiently we have developed the Genetic Output Analysis Tool (GOAT) to improve visualization and annotation of GWAS data. GOAT offers interactive search of GWAS datasets via specific queries to identify significant associations between multiple SNPs and phenotypes. GOAT was designed to be scalable and operates on top of "Big Data" technologies. It is programmed in python and can be connected directly to any database using an Apache server. This paper describes some of the GOAT's leading features and characteristics and compares them to existing open source GWAS visualization tools. We also present future development plans.**

Keywords—Bioinformatics; GOAT; GWAS; Genomic region; Visualization tool, Software Engineering, Spark, Big Data.

I. Introduction

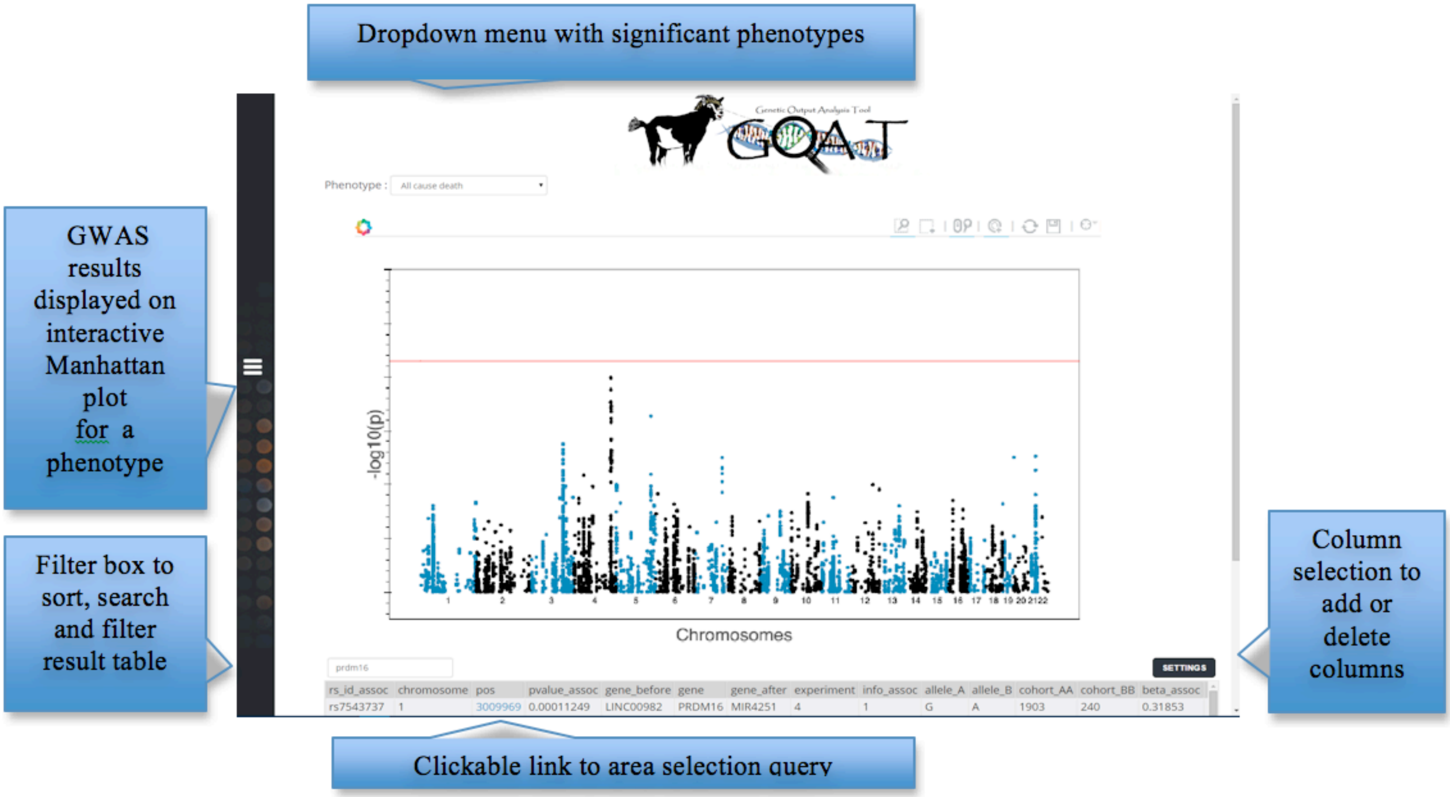
Large scale genome wide genotyping studies has offered researchers the opportunity to map genes of complex diseases such as diabetes [1], and chronic kidney diseases [2] which are known to be polygenic, and multi-factorial [3]. Genome Wide Association Studies (GWAS) are a widely utilized method applied to large genetic datasets to identify genetic variants associated with specific diseases or phenotypes and their associated complications [4]. Output from these types of studies involves the analysis of millions of single nucleotide polymorphisms (also called SNPs) which are examined at the same time and classified by their strength of association with the disease. GWAS results are initially visualized using Manhattan plots where the –log of the associated SNP p-values are plotted against physical map position of SNPs in relation to their chromosomal position within the genome. A unique GWAS, that can generate millions of data points, is produced for each phenotype resulting in complex datasets. It is instrumental that researchers have the ability to query all this complex data interactively across multiple phenotypes even though most of the associations do not reach a nominal significance threshold [5]. Complex diseases involve SNPs responsible for the disease that will be in a non-coding area of the DNA, or will be in linkage desequilibrium (LD) with the most significant SNPs, so it's very important to capture all SNPs surrounding those areas [3]. Visualization tools help geneticists to capture "ALL" relevant data, to better understand complex "OMICS" data at multiple levels and to generate novel pathogenetic hypotheses. Furthermore, these datasets are usually stored in a relational database and, for the average genomic researcher, it is challenging to effectively query this large amount of data. These considerations constituted our impetus to design the Genetic Output Analysis Tool (GOAT).

The next sections summarize the GOAT's user requirements, present its key functionalities, internal structure and previews future improvements planned.

II. GOAT's key features

II A. GOAT users requirements

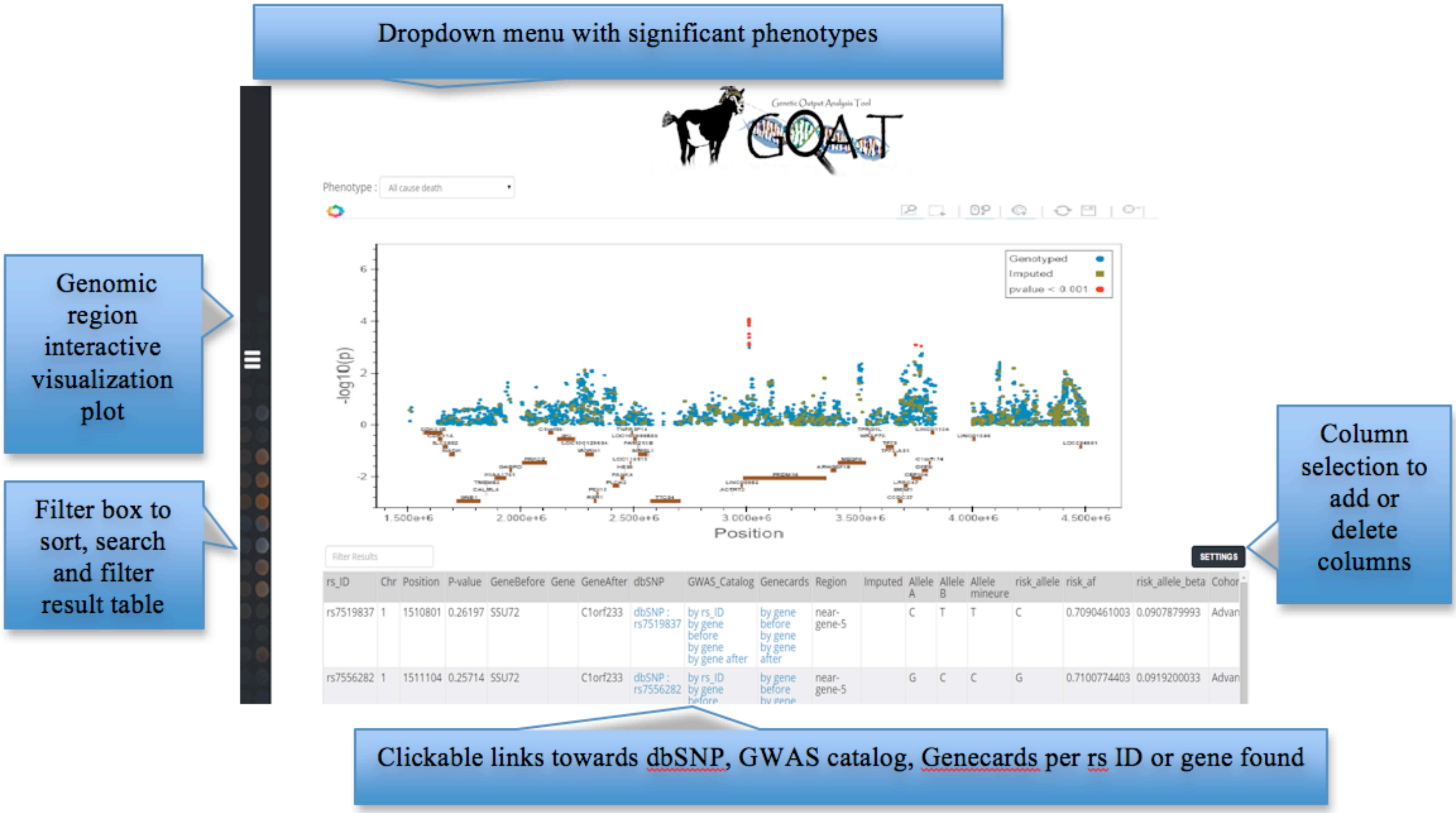
Our first task consisted of building an interface that allowed a query of "ALL" data by gene or by SNP rs identification number to view a genomic region associated with a phenotype and to allow interpretation of the data in an efficient manner.


Dropdown menu with significant phenotypes
Genetic Output Analysis Tool
GOAT
Phenotype : All cause death
-log10(p)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
Chromosomes
GWAS results displayed on interactive Manhattan plot for a phenotype
Filter box to sort, search and filter result table
Column selection to add or delete columns
Clickable link to area selection query
SETTINGS

prdm16

rs_id_assoc	chromosome	pos	pvalue_assoc	gene_before	gene	gene_after	experiment	info_assoc	allele_A	allele_B	cohort_AA	cohort_BB	beta_assoc
rs7543737	1	3009969	0.00011249	LINC00982	PRDM16	MIR4251	4	1	G	A	1903	240	0.31853

Figure 1. Result page generated after query by gene or rs ID with a selected threshold of 0.001.


Dropdown menu with significant phenotypes
Genetic Output Analysis Tool
GOAT
Phenotype : All cause death
Genotyped
Imputed
pvalue < 0.001
-log10(p)
6
4
2
0
-2
1.500e+6
2.000e+6
2.500e+6
3.000e+6
3.500e+6
4.000e+6
4.500e+6
Position
PRDM16
Genomic region interactive visualization plot
Filter box to sort, search and filter result table
Column selection to add or delete columns
Clickable links towards dbSNP, GWAS catalog, Genecards per rs ID or gene found
SETTINGS

Filter Results

rs_ID	Chr	Position	P-value	GeneBefore	Gene	GeneAfter	dbSNP	GWAS_Catalog	Genecards	Region	Imputed	Allele A	Allele B	Allele mineure	risk_allele	risk_af	risk_allele_beta	Cohor
rs7519837	1	1510801	0.26197	SSU72		C1orf233	dbSNP : rs7519837	by rs_ID by gene before by gene by gene after	by gene before by gene by gene after	near-gene-5		C	T	T	C	0.7090461003	0.0907879993	Advan
rs7556282	1	1511104	0.25714	SSU72		C1orf233	dbSNP : rs7556282	by rs_ID by gene before	by gene before by gene	near-gene-5		G	C	C	G	0.7100774403	0.0919200033	Advan

Figure 2. Area selection result page generated by selecting position 64 000 000 and chromosome 1 and taking an interval of +/- 1 500 kbp from that position.

A requirement document (named a vision document) was initially created to capture all prerequisites aimed at facilitating the analytical flow. This was done by interviews conducted with members of a multidisciplinary research team that included geneticists, clinicians, bio-statisticians and bio-informaticians. A user-friendly query system that allowed for quick data mining, visualization and genomic analysis was highly anticipated by the researchers. A requirement was the need for interactive access to graphical representations of the data accompanied by the possibility to download snapshots of the graphs for further discussions, presentations and publications. Another higly desired feature of this software was that all table results contain selectable columns with links to external databases, such as GWAS catalog [6], Genecards [7] and dbSNP [8] for accurate referencing.

The GWAS catalog is a quality controlled, manually curated and literature-driven collection of GWAS that is resulting from a collaboration between EMBL-EBI [9] and GWAS study assays. It reports GWAS data that includes at least 100 000 SNPs and for which SNP-trait associations have p-values in the order of $< 10^{-5}$. Genecards is also a searchable, integrated database of human genes that provides comprehensive, updated, and user-friendly information on all known and predicted human genes. It includes links and informations about other databases such as Ensembl, Uniprot, OMIM [10], NCBI [11], ClinVar [12] and Pubmed [13]. It extracts and integrates gene-related data, including genomic, transcriptomic, proteomic, genetic, clinical and functional information, which are automatically mined from more than 100 carefully selected web sources, thereby allowing integrated access to all this information overcoming problems of data formatting and heterogeneity. It also uses standard nomenclature and approved gene symbols. Finally, GOAT showa also dbSNP informations. dbSNP is a free public archive for genetic variation within and across different species developped and hosted by NCBI.

II B. GOAT 'S FIRST VERSION

FUNCTIONALITIES

The first version of GOAT (still a prototype) was designed in a way to enable researchers to query their database by searching by gene name or SNP rs ID and by offering the flexibility to select a display threshold. In this sense, GOAT is designed to give researchers the flexibility and independence to query multiple GWAS without requiring a bioinformatician expert's help.

The idea behind this interactive interface is to enable researcher's who don't have the appropriate programming skills or database knowledge to search for the most interesting results, nor the knowledge about bioinformatics file formats while at the same time, generate visual tools adapted to the genetic research domain to support their hypothesis. GOAT achieves this by selecting the most interesting SNPs and phenotypes, above a 0.001 threshold on the p-value, from its database and generates a result page containing an interactive Manhattan plot and a table result, where all significant SNPs are ordered from the most significant to the least significant (see Fig. 1).

An interesting feature, added during testing, is a dropdown menu where all the phenotypes for which SNPs or genes were found significant are also available for visualization, by filtering them from the most significant to the least significant. A table of results is also displayed under the graph which is interactive allowing filtration, sorting, and column selection or masked for display, so that all the relevant information needed for analysis and interpretation can be displayed and extracted. The interactiveness of the Manhattan plot enables the researcher for fast identification of interesting peaks providing information on rs number, gene name and p-values in a tag.

The information displayed in this table are: the covariates used with the phenotype analyzed, the chromosome, the SNP position, the p-values that meet the selected threshold value of significance, the gene in which the SNP is located, its closest upstream and downstream genes, as well as other key informations such as allele identification, allelic frequency in the population used, beta, odds ratio, hetOR, cases and controls that are identified with their alleles and frequencies. In other words this first result page allows the researcher to have an overview of the GWAS results.

Any genomic region displayed in the Manhattan plot can be zoomed, by clicking on a SNP displayed in the table, within a +/- 1500 kbp interval from that SNP and displayed in a second result page named "Area selection" (see Fig. 2).

The graph shown in figure 2 is interactive and is designed in a way that it allows easy navigation through it. It offers key features like: 3D views that show where the SNPs are mostly concentrated, zooming, mouse over on any data point and displaying rsID number, the gene in which it is, the p-value obtained in the GWAS, the region (i.e. exon or intron), and finally this graph can be downloaded at any stage of the zooming in a publication ready format.

Note hat the "Area selection" page includes all SNPs present in the interval whether significantly associated or not, with links to the aforementioned external databases.

The source code and vision document for this first version are available on Github by looking for GOAT_Genetic_Ouput_Analysis_Tool.

III. COMPARING GOAT TO OTHER OPEN-SOURCE SOFTWARES

GOAT's main features were compared to two popular alternative GWAS visualization tools, namely LocusZoom [14] and the Integrative Genomics Viewer of the Broad Institute, Boston USA (IGV viewer) [15].

III A. COMPARING GOAT TO LOCUSZOOM

LocusZoom is a web-based plotting open-source software that provides visual display of GWAS results in a publication ready format. The region to be displayed is specified using a pull-down menu, but the user has to select the SNP IDs and P-values to be displayed. Typical run time for a single plot returned to the browser window is ~20 seconds for an interval of 200kbp and it is a pdf file. In comparison, our tests show that GOAT's area selection graph responds within ~5 seconds for an area of 3000kpb and it is selected directly from the general view containing the Manhattan plot (refer to Fig. 1). The interval is selected from the database where the name and location of genes in the UCSC Genome Browser hg19 model has been preloaded. The graph is then available in a publication ready format. It is also fully interactive and can be downloaded as a PNG (portable network graphics) file from any zooming.

III B. Comparing GOAT to IGV viewer

The other open source visualization software that we compared to GOAT is the IGV viewer. IGV is a desktop application developed to support a diverse range of data types including: Next-Generation Sequencing and array-based platforms, such as expression and copy-number arrays. It is programmed in Java and runs on mamy platforms, such as, Windows, Mac OS X and Linux. The plots are interactive and large scale datasets can be explored on a desktop computer. It supports file formats such as gff, bed, wig, bam, tdf and allows session saving and plots sharing along with data file sharing.

The main differences between GOAT and IGV viewer are that GOAT provides graphs in a publication ready quality format, offers a 3D view, and that contrary to IGV, GWAS results do not need any pre-formatting as all the information needed to generate the graphs is downloaded directly from the preloaded library stored in the database.

A summary table highligting GOAT's features was produced (See Table 1). We compared performance, efficiency and the ability to produce publication ready plots. While these tools have different functionalities, very modern features were implemented in GOAT in order to create a whole new visualisation experience that, we hope, will satisfy even the most advanced genomic researcher.

Table 1. Summary of results after comparison of GOAT's performance, efficiency and the ability to produce publication ready plots to visualize genomic regions, with commonly used tools such as LocusZoom and IGV.

Plots comparison	LocusZoom	IGV	GOAT
Genomic interval	+/- 200 kbp	+/- 124 mbp	+/- 1.5 mbp
Zooming	No	Yes	Yes
3D view	No	No	Yes
SNP indentification	Yes	Yes	Yes
Interactive	No	Yes	Yes
Publication ready	Yes	No	Yes
Time to generate	~20 seconds	~ 1 second	~ 5 seconds
Prior data formatting	Yes	Yes	No

IV. GOAT's Software Design

IV A. Technologies used for GOAT's User Interface

For the user interface (i.e. front-end), a Javascript framework, developed by Facebook under an open source BSD (Berkeley Software Distribution License), named React [16] was used. This framework was initially developed to solve the challenges encountered when interacting with complex user interfaces involving datasets that change over time. React solves this problem by introducing a new paradigm and allows scalable and maintainable Javascript applications and user interfaces. This open source technology allowed us to use components such as Griddle, created by Ryan Lanciaux, to easily render interactive tables. For the rest of the user interface, we used SASS [17], one of the most mature, stable and powerful open source CSS preprocessors available at the present time. All these technologies allowed for a modular design of GOAT's front-end. Finally, for a seemless integration, we used SMACSS categorizarion features to ensure easy readability, scalability and future maintenance and of the resulting CSS source code [18].

IV B. Frameworks used for GOAT's core functionalities

For the main functionalities of GOAT, the Django framework was deployed to manage the web interface and the Python back-end code. Python is popular with bioinformaticians [19]. In this first version of GOAT we use a MySQL relational database. MySQL will not survive long in a BigData environment and we already plan to replace it by Spark RDD's in the next release. All genomic datasets used originate from the Advance trials which is one of the largest controlled and randomized trial ever to be conducted on type 2 diabetes patients [20]. These datasets were queried using an SQL query and then integrated in the python code where 1) a Pandas dataframe; and 2) a numpy library were used to manipulate the data before rendering the graphs. The Manhattan plot is rendered using Biopython's Matplotlib, from Continuum Analytics, which is also widely used in the health research area for rendering static graphs [21].

The interactive graph was generated using the Bokeh library from Continuum Analytics. This recent open-source technology has interactive capabilities that target the presentation of streaming data on a browser and interactively allow for the visualization of very large datasets. It allows D3.js style graphics, but uses canvas instead of SGV (Stacked Graph Visualization) [22]. It allows to render more objects at a time [23]. Finally sorting and handling of the data were done using: 1) Numpy, an extension of the Python programming language; and 2) Pandas, an open-source library with high performance easy to use data structure and ColumnDataSource which is a Bokeh object as a Python interactive visualization library.

Conclusion

GOAT's key functionalities gives genomic researchers a user friendly user interface and gives them the ability to be autonomous during their GWAS discovery iterations. It also generates publication ready Manhattan graphs, as well as allow them to interactively data mine massive amount of genetic data.In future releases of GOAT, the Manhattan plot will be made interactive, and the user will be able to do phenotype comparisons, view Linkage disequilibrium (LD), and recombination rate for results dislayed at the area selection page. As well we plan to add a statistical analysis module that will include popular algorithms used frequently in genetics research such as population distribution, survival analysis, association, correlation, interaction, principal components analysis and recent machine learning techniques.

The performance and scalability of GOAT will also be improved by replacing the MySQL back-end with a Spark back-end. The data will be structured reusing the existing Python programs but through a Resilient Distributed Dataset (RDD) which allows for a distributed memory abstractions for in-memory computations on large fault-tolerent clusters [24]. This recent and powerful Big-Data technology will be tested further at the CRCHUM.

Acknowledgment

We would like to thank Olivier Mirandette for his advice on Python and Bokeh server performance issues and PhD student David Lauzon for early work on the Django architecture.

References

[1] Sladek R, Rocheleau G, Rung J, Dina C, Shen L, Serre D, et al. A genome-wide association study identifies novel risk loci for type 2 diabetes. Nature. 2007 Feb 22;445(7130):881-5.

[2] Teumer A, Tin A, Sorice R, Gorski M, Yeo NC, Chu AY, et al. Genome-wide association studies identify genetic loci associated with albuminuria in diabetes. Diabetes. 2015 Dec 2:db151313.

[3] Grossman E, Messerli FH. Hypertension and diabetes. Advances in cardiology. 2008;45:82.

[4] Londin E, Yadav P, Surrey S, Kricka LJ, Fortina P. Use of linkage analysis, genome-wide association studies, and next-generation sequencing in the identification of disease-causing mutations. In Pharmacogenomics 2013 Jan 1 (pp. 127-146) Humana Press.

[5] Balding DJ. A tutorial on statistical methods for population association studies. Nat. Rev. Genet., 2006 Oct 1;7(10):781-91.

[6] Hindorff LA, Junkins HA, Hall PN, Mehta JP, Manolio TA. A catalog of published genome-wide association studies. National Human Genome Research Institute. 2011 Apr.

[7] Rebhan M, Chalifa-Caspi V, Prilusky J, Lancet D. GeneCards: integrating information about genes, proteins and diseases. Trends in Genetics. 1997 Apr 30;13(4):163.

[8] Kitts A, Sherry S. The single nucleotide polymorphism database (dbSNP) of nucleotide sequence variation. The NCBI Handbook. McEntyre J, Ostell J, eds. Bethesda, MD: US National Center for Biotechnology Information. 2002.

[9] Goujon M, McWilliam H, Li W, Valentin F, Squizzato S, Paern J, Lopez R. A new bioinformatics analysis tools framework at EMBL–EBI. Nucleic acids research. 2010 Jul 1;38(suppl 2):W695-9.

[10] Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research. 2005 Jan 1;33(suppl 1):D514-7.

[11] Coordinators NR, Acland A, Agarwala R, Barrett T, Beck J, Benson DA, Bollin C, Bolton E, Bryant SH, Canese K, Church DM. Database resources of the national center for biotechnology information. Nucleic acids research. 2014 Jan;42(Database issue):D7.

[12] Landrum MJ, Lee JM, Riley GR, Jang W, Rubinstein WS, Church DM, Maglott DR. ClinVar: public archive of relationships among sequence variation and human phenotype. Nucleic acids research. 2014 Jan 1;42(D1):D980-5.

[13] McEntyre J, Lipman D. PubMed: bridging the information gap. Canadian Medical Association Journal. 2001 May 1;164(9):1317-9.

[14] Pruim RJ, Welch RP, Sanna S, Teslovich TM, Chines PS, Gliedt TP, et al. LocusZoom: regional visualization of genome-wide association scan results. Bioinformatics. 2010 Sep 15;26(18):2336-7.

[15] Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in bioinformatics. 2012 Apr 19:bbs017.

[16] Gackenheimer C. What Is React?. InIntroduction to React 2015 (pp. 1-20). Apress.

[17] Prabhu A. Introduction to Preprocessors. InBeginning CSS Preprocessors 2015 (pp. 1-12). Apress.

[18] Jain N. Review of different responsive CSS Front-End Frameworks. Journal of Global Research in Computer Science. 2015 Apr 2;5(11):5-10.

[19] Moore D, Budd R, Wright W. Professional Python Frameworks: Web 2.0 Programming with Django and Turbogears. John Wiley & Sons; 2008 Jan 22.

[20] Patel A, ADVANCE Collaborative Group. Effects of a fixed combination of perindopril and indapamide on macrovascular and microvascular outcomes in patients with type 2 diabetes mellitus (the ADVANCE trial): a randomised controlled trial. The Lancet. 2007 Sep 14;370(9590):829-40.

[21] Hunter JD. Matplotlib: A 2D graphics environment. Computing in science and engineering. 2007 May 1;9(3):90-5.

[22] Bokeh. 2015. Available at:http://bokeh.pydata.org/en/latest/.

[23] Barnard L, Mertik M. Usability of Visualization Libraries for Web Browsers for Use in Scientific Analysis. Int. Jrnl. Comp. App. 2015 Jan 1;121(1).

[24] Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. InProceedings of the 9th USENIX conference on Networked Systems Design and Implementation 2012 Apr 25 (pp. 2-2). USENIX Association.