An open-source platform to distribute and interpret data from multiplexed assays of variant effect

Daniel Esposito[1, †], Jochen Weile[2, 3, 4, 5, †], Jay Shendure[6, 7, 8], Lea M Starita[6, 7], Anthony T Papenfuss[1, 9, 10, 11, 12], Frederick P Roth[2, 3, 4, 5, 13, *], Douglas M Fowler[6, 13, 14, *], Alan F Rubin[1, 9, 10, *]

[1]Bioinformatics Division, The Walter and Eliza Hall Institute of Medical Research, Parkville, VIC, Australia
[2]The Donnelly Centre, University of Toronto, Toronto, ON, Canada
[3]Lunenfeld-Tanenbaum Research Institute, Sinai Health System, Toronto, ON, Canada
[4]Department of Molecular Genetics, University of Toronto, Toronto, ON, Canada
[5]Department of Computer Science, University of Toronto, Toronto, ON, Canada
[6]Department of Genome Sciences, University of Washington, Seattle, WA, USA
[7]Brotman Baty Institute for Precision Medicine, Seattle, WA, USA
[8]Howard Hughes Medical Institute, University of Washington, Seattle, WA, USA
[9]Department of Medical Biology, University of Melbourne, Melbourne, VIC, Australia
[10]Bioinformatics and Cancer Genomics Laboratory, Peter MacCallum Cancer Centre, Melbourne, VIC, Australia
[11]Sir Peter MacCallum Department of Oncology, University of Melbourne, Melbourne, VIC, Australia
[12]Department of Mathematics and Statistics, University of Melbourne, Melbourne, VIC, Australia
[13]Canadian Institute for Advanced Research, Toronto, ON, Canada
[14]Department of Bioengineering, University of Washington, Seattle, WA, USA

[†]Authors contributed equally
*Correspondence to alan.rubin@wehi.edu.au , dfowler@uw.edu, fritz.roth@utoronto.ca

Abstract

Multiplex Assays of Variant Effect (MAVEs), such as deep mutational scans and massively parallel reporter assays, test thousands of sequence variants in a single experiment. Despite the importance of MAVE data for basic and clinical research, there is no standard resource for their discovery and distribution. Here we present MaveDB, a public repository for large-scale measurements of sequence variant impact, designed for interoperability with applications to interpret these datasets. We also describe the first of these applications, MaveVis, which retrieves, visualizes, and contextualizes variant effect maps. Together, the database and applications will empower the community to mine these powerful datasets.

Keywords

Deep mutational scanning, massively parallel reporter assays, large-scale mutagenesis, MAVE, multiplexed assay of variant effect, genome interpretation, personalized medicine

Background

Experimentally interrogating the effects of genetic variation has helped reveal the mechanisms by which genes function and facilitate an understanding of the clinical consequences of human genetic variation. Multiplex Assays of Variant Effect (MAVEs) leverage high-throughput DNA sequencing to greatly increase the scale at which variants can be experimentally investigated [1–3]. A MAVE yields a set of scores that describe the functional effect of thousands to tens of thousands of variants of a coding sequence, promoter, enhancer or other genetic element relative to a reference sequence. MAVEs are being adopted rapidly for both basic research and clinical applications [4]. As a consequence, the total number of variants with functional data generated by MAVEs was predicted to surpass 200,000 by the end of 2018 [3], which exceeds the number of classified missense variants available in ClinVar [5].

These large-scale variant effect maps are yielding insights into protein function, structure, and evolution [6–10]; exploring gene regulation and promoter function [11–13]; improving computational variant effect prediction [14, 15]; and guiding variant interpretation in the clinic [16–20]. However, the impact of variant effect maps has been limited by shortcomings in data availability, dissemination, and discoverability. Many publications describing large-scale variant effect mapping do not provide variant effect scores for all variants that were assayed. When variant effect scores are provided, they are often accessible only as a supplementary table or via a bespoke web interface [18, 19, 21] leading to a proliferation of inconsistent formats. Some publications, instead of including variant effect scores, deposit the associated high-throughput DNA sequencing data in the Sequence Read Archive or Gene Expression Omnibus [22, 23]. This raw data can be used to reconstruct variant effect scores, but accurately replicating the original analysis can be non-trivial. While databases of variant effect information exist, they are typically designed for a specific application [24–26] or a specific group of target genes [27–29]. Larger and more general databases can sometimes contain variant effect data [30, 31], but these platforms were not developed with large-scale variant effect maps in mind, so valuable context for the variant effect scores and associated metadata may be lost. Furthermore, most existing resources lack support for noncoding targets entirely.

To overcome these challenges and facilitate future advances, we are establishing an open-source platform for MAVE resources. The foundation is MaveDB, a central repository that allows researchers to store and publish processed MAVE datasets, associated metadata, and linked raw data using a machine readable, standardized, and searchable format. An easy-to-use web interface maximizes the impact and usefulness of researchers' work by making data readily accessible to the whole community, whether for clinical applications, meta-analysis, or reanalysis as computational techniques are refined.

This platform is designed to allow additional applications to communicate directly with MaveDB. The first of potentially many such applications, MaveVis, visualizes and provides context to protein variant effect maps by generating heatmaps and automatically integrating them with secondary structure, surface accessibility, interaction interfaces, and conservation data.

Organization and content of MaveDB

To capture the structure of real-world study designs, MaveDB is organized hierarchically into score sets, experiments, and experiment sets (**Figure 1**). Score sets, the most basic unit of organization, contain the variant effect scores and additional metadata such as target sequence information and detailed methods. Each variant effect score is a numeric value. Optional data columns containing values related to each variant effect score such as variant counts and measures of uncertainty can also be included and named by the user.

Most experimental designs in MaveDB involve multiple score sets. For example, protein MAVEs commonly have one score set for nucleotide variants and another for amino acid variants [32]. Experiments with tiled designs [33, 34] can have score sets for each tile, and experiments with multiple distinct reference sequences [35] can have score sets for each reference sequence [35][40]. In addition, we envision that reanalysis and renormalization of existing datasets using updated methods will be commonplace [14, 15, 36, 37]. By grouping all analyses of a single raw dataset under one experiment, MaveDB ensures that the number of assays performed on each target sequence can be tracked accurately.

Each experiment describes one or more analyses arising from a single MAVE, including any technical and biological replicates. In addition to links to score sets, experiments contain metadata including methodological details, links to raw data, and associated publications (**Table 1**), but no variant score information. Experiment sets contain one or more related experiments, for example multiple MAVEs performed on the same target sequence under different conditions or multiple experiments from the same publication.

MaveDB currently contains over one million variant effect scores across 39 unique targets. We welcome both new and previously described datasets from the community and have implemented a conversion tool, mavedb-convert, for datasets generated by Enrich [38], Enrich2 [37], and EMPIRIC [39] (see Availability section).

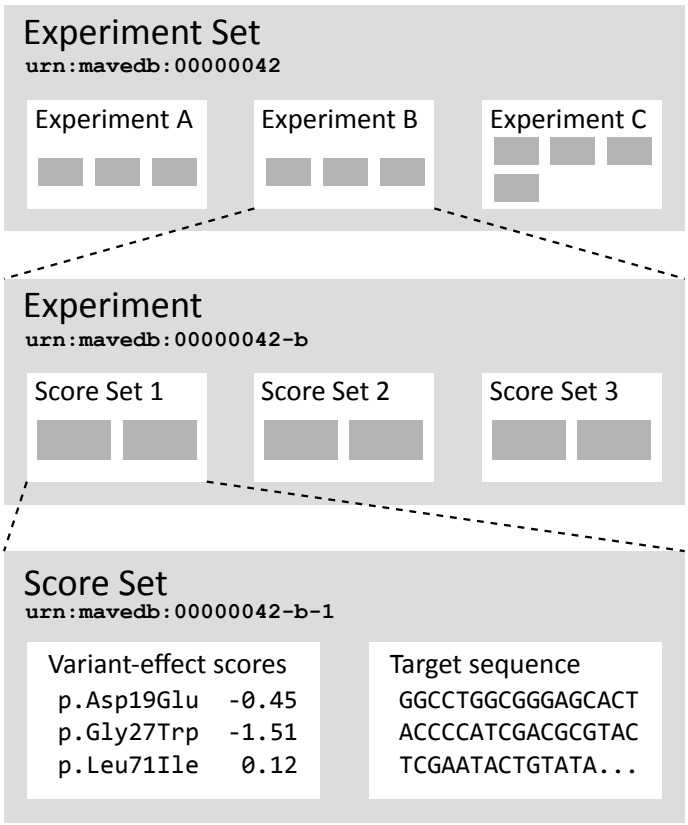

Figure 1: Relationships between MaveDB entries. MaveDB has three main entry types arranged in a hierarchical structure. The URN accession numbers shown for each example entity reflect these relationships. Additional metadata fields for score set and experiment entries are listed in **Table 1**.

Utility of MaveDB

Accessing datasets

MaveDB can be accessed through a standard web browser that allows users to explore by keyword, target gene, or organism. Alternatively, the advanced search function allows users to query all metadata fields, including the full text of methods and abstracts. Complete sets of variant effect scores and related values can be downloaded from any score set page in comma-separated value format. These files can be parsed easily in most scientific programming environments or imported into spreadsheet applications.

The advanced search function is also accessible programmatically through the REST API (Representational State Transfer–Application Programming Interface). The API returns structured data, including full score sets and metadata, in JSON format, suitable for deserialization by most programming languages. Users of the R programming environment [40] can access MaveDB's REST API using the rapimave library, which also includes a suite of exploration, searching, parsing and filtering functions (see Availability section).

Creating new entries

Typically, a user starts by creating an experiment. The experiment can be added to an existing experiment set if desired, otherwise a new one will be created automatically. The user provides a description of the assay used to generate the raw data, adds links to the raw data if available, and can then add contributors. After the experiment is created, the user

creates one or more associated score sets. Here, the user enters required information about the target such as its name and sequence, and also describes the methods used to calculate the variant effect scores from the raw data. Variant effect scores and optional counts files are then uploaded via the web interface and validated by the server.

Publishing datasets

When first created, score sets, experiments, and experiment sets are private and have temporary accession numbers. Private entries are only viewable by their contributors and all values may be modified. Private entries can be accessed through the API by providing a contributor's private access token generated on the contributor's user profile page.

Completed private score sets can be published, making the score set publicly viewable. Publication creates a stable accession number and freezes the target sequence and variant effect score data, ensuring that all subsequent analyses based on the data are recomputable. Associated experiment and experiment sets are also published automatically if they are still private. Users may continue to edit some metadata such as the methods and description after publication.

Published scores cannot be changed, but in case a correction is necessary, MaveDB allows score sets to be deprecated when creating a replacement. Users browsing MaveDB will only see the most recent version, but deprecated score sets will remain available by accession number to ensure that previous analyses are reproducible.

Contributor permissions

MaveDB supports three contributor roles: administrator, editor and viewer. Administrators can add or remove contributors, modify entries, and publish score sets. Editors can modify entries but cannot affect the contributor list or make entries public. Viewers can see their private entries in the database but cannot change them.

All three roles appear in the contributor list with no visual distinction between them, and administrators can continue to change the contributor list for each score set or experiment after publication. Since score sets and experiments have independent contributor lists, MaveDB maintains clear attribution when datasets are reanalyzed.

Data licensing

Administrators may select one of several Creative Commons licenses for each score set [41–43] and additional licensing options may be added in response to user requests. The license information is included as score set metadata and as part of the header of each downloaded file. Administrators can relicense after publication, although users who download under a more permissive license would not be subject to a more restrictive license.

Table 1: MaveDB metadata fields

Field name	Experiment	Score Set	Type	Searchable	Link
Keyword	✓	✓	String	✓	
Abstract	✓	✓	Markdown	✓	
Method	✓	✓	Markdown	✓	
Short description	✓	✓	String	✓	
Title	✓	✓	String	✓	
PubMed ID	✓	✓	Accession	✓	✓
DOI	✓	✓	Accession	✓	✓
SRA accession	✓		Accession	✓	✓
RefSeq accession		✓	Accession	✓	✓
Ensembl accession		✓	Accession	✓	✓
UniProt accession		✓	Accession	✓	✓
Created by	✓	✓	Contributor	✓	✓
Last modified by	✓	✓	Contributor	✓	✓
Creation date	✓	✓	DateTime	✓	
Modification date	✓	✓	DateTime	✓	
Publication date	✓	✓	DateTime	✓	
Licence		✓	Licence	✓	✓
Has replacement		✓	Boolean	✓	

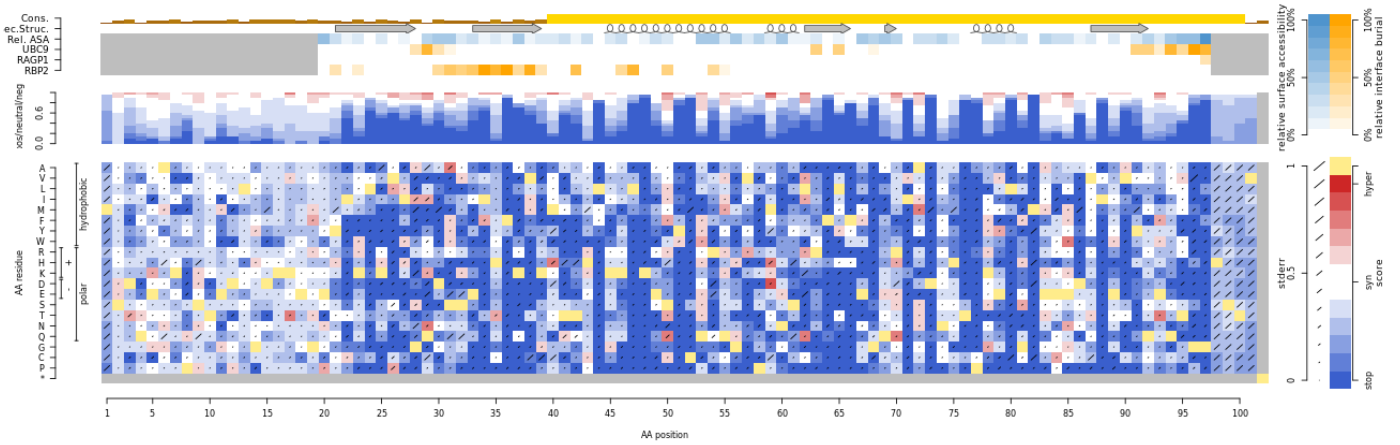

Figure 2: Heatmap for the SUMO1 MAVE dataset rendered by MaveVis. The x-axis iterates over amino acid positions in the protein, while the y-axis lists all possible amino acid changes organized by their physicochemical properties. The heatmap color reflects the variant effect score, with blue being as deleterious as a full deletion, white being equivalent to the reference allele, and red representing a stronger phenotype than the reference residue at that position. Yellow cells indicate the reference amino acid at each position. Error bars represent standard error of the mean. The stacked bars above the heatmap represent the relative frequencies for each phenotype bin of corresponding color at each position. Additional tracks show data integrated from other databases: orange heatmap tracks represent burial in protein interaction interfaces, the steel blue heatmap track represents solvent accessibility, the arrows and spirals correspond to secondary structure, and the yellow bar chart at the top shows sequence conservation.

Visualizing variant effect maps with MaveVis

The MaveVis application allows users to quickly visualize score sets retrieved directly from MaveDB. One example of MaveVis output for a variant effect map of the protein SUMO1 [44] is shown in **Figure 2**. Score sets are rendered as heatmaps with additional tracks representing integrated structural and conservation information from PDB [45] and UniprotKB [31]. The heatmap shows all possible amino acid changes at each protein sequence position, with colors reflecting the variant effect scores. The color scale is automatically calibrated based on the scores of reference and null alleles in the dataset or set manually by the user. Error bars are drawn directly on the heatmap fields to represent the measurement error provided in the score set, if present. Additional tracks show burial in protein interaction interfaces, residue-specific solvent accessibility, protein secondary structure, and sequence conservation.

Accessing MaveVis

MaveVis is hosted at http://varianteffect.org, a portal for applications built on MaveDB. Users can follow the MaveVis link on each MaveDB score set page or navigate directly to http://vis.varianteffect.org and search for datasets. Once a score set is selected, the corresponding UniProt accession from MaveDB is suggested when available. MaveVis automatically presents potentially relevant PDB structures for the selected protein that overlap with the score set target sequence, allowing users to select which structures to include in the visualization. The resulting plot can be downloaded in PNG, PDF or SVG format.

In addition to the web interface, MaveVis also exists as an R package for local use (see Availability section). The R package provides direct access to both the visualization and underlying data integration functions, making it easy to automatically compile structural and conservation feature tables for individual proteins.

Interaction with MaveDB

The MaveVis server automatically synchronizes with MaveDB at regular intervals via its API, caching any new score sets, automatically obtaining relevant PDB and UniProt data, and pre-calculating partial results for a more responsive user experience. MaveVis also exposes its own API, allowing it to be used within more complex workflows.

To facilitate communication between MaveVis and MaveDB, we developed an R package, hgvsParseR, to parse or assemble HGVS [46] strings that describe alleles (see Availability section). In addition to its utility for visualizing variant effect maps, we expect that this package will be generally useful for working with data from ClinVar [5], gnomAD [47], and other important sequence variation resources.

Conclusions

MaveDB is the foundation of an open-source platform for the collection, distribution, and analysis of variant effect maps. Designed to be flexible and extensible, the MaveDB repository can accommodate data from diverse target sequences and experimental methods as the field evolves. Using MaveDB to combine variant effect data with external contextual information, MaveVis is the first application built on this resource. We envision developing additional applications such as tertiary structure analysis, automatic imputation of missing values in variant effect maps [48], and a broadly-applicable dashboard to assist dataset interpretation.

MaveDB, MaveVis, and Enrich2 simplify, standardize, and democratize MAVE data analysis. These tools are the beginnings of a community-driven, open-source platform that allows researchers to explore these comprehensive datasets. The impact of each dataset will continue to increase as the number of assayed variants grows, contributing to a more complete understanding of genetic variation and sequence function.

Methods and implementation

MaveDB is implemented in Python using the Django Python Web framework [49, 50]. The relational database backend is PostgreSQL [51]. Asynchronous tasks such as handling file uploads and sending emails are managed using RabbitMQ and Celery [52, 53]. Variant score and count data are stored using PostgreSQL JSONField objects, which offer additional flexibility for storing arbitrarily-named data columns compared to a more traditional relational database design. Database accession numbers for publicly accessible entries are assigned in URN (Universal Resource Name) format [54] and their structure reflects MaveDB's hierarchy (**Figure 1**).

Differences between each variant sequence and the target sequence are described using HGVS format [46]. MaveDB supports variant strings that describe substitutions or small indels in most sequence contexts.

Contributors are authenticated using their ORCID iD via the OAuth2 protocol [55, 56]. Consequently, an individual must have an ORCID iD to be named as a contributor to a MaveDB dataset. Users do not need to log in to browse or download publicly available data. MaveDB allows users to provide a private contact email address if they want to be contacted by administrators or receive alerts, but all other details are pulled from their public ORCID record.

Abstract and methods sections support Markdown [57] blocks for formatted text with support for mathematical notation. Markdown blocks are rendered to HTML using Pandoc [58].

MaveVis is implemented using R [40] and Docker [59]. Surface accessibility and interface burial are calculated using FreeSasa [60]. Secondary Structure is calculated using DSSP [61]. Conservation tracks are calculated using the AMAS algorithm [62], based on multiple alignments computed using ClustalOmega [63] for the appropriate UniRef90 set of orthologous proteins with at least 90% sequence identity from UniProtKB [31].

Declarations

Availability of data and material

MaveDB is hosted at https://www.mavedb.org/

MaveVis is hosted at http://vis.varianteffect.org/

Source code for all websites, tools, and packages is available on GitHub at https://github.com/VariantEffect

- MaveDB: https://github.com/VariantEffect/mavedb
- MaveVis: https://github.com/VariantEffect/mavevis
- mavedb-convert: https://github.com/VariantEffect/mavedb-convert
- rapimave: https://github.com/VariantEffect/rapimave
- hgvsParseR: https://github.com/VariantEffect/hgvsParseR

A pre-compiled docker image for MaveVis is also available on DockerHub at https://hub.docker.com/r/jweile/mavevis/

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by the National Institutes of Health (NIH; R01GM109110 to DMF) and the Brotman Baty Institute for Precision Medicine. D.M.F. is a CIFAR Azrieli Global Scholar. A.T.P. was supported by the Lorenzo and Pamela Galli Charitable Trust and by an Australian National Health and Medical Research Council (NHMRC) Program Grant (1054618) and NHMRC Senior Research Fellowship (1116955). The research benefitted by support from the Victorian State Government Operational Infrastructure Support and Australian Government NHMRC Independent Research Institute Infrastructure Support. F.P.R. and J.W. gratefully acknowledge funding by the One Brave Idea Initiative, the National Human Genome Research Institute of the NIH Center of Excellence in Genomic Science Initiative (HG004233), the Canadian Excellence Research Chairs Program, and a Canadian Institutes of Health Research Foundation Grant.

Authors' contributions

A.F.R., D.M.F., and F.P.R. conceived of the project. D.E. and A.F.R. built the database and its web interface. J.W. built the MaveVis application. D.E. and J.W. built the APIs. All authors wrote the manuscript and approved the final version.

Acknowledgements

We would like to acknowledge Bernie Pope, Peter Georgeson, Nick Moore, Matthew Wakefield, and Dan Bolon for helpful advice and guidance.

References

1. Fowler DM, Fields S. Deep mutational scanning: a new style of protein science. Nat Methods. 2014;11:801–7.

2. Gasperini M, Starita L, Shendure J. The power of multiplexed functional analysis of genetic variants. Nat Protoc. 2016;11:1782–7.

3. Weile J, Roth FP. Multiplexed assays of variant effects contribute to a growing genotype–phenotype atlas. Hum Genet. 2018. doi:10.1007/s00439-018-1916-x.

4. Starita LM, Ahituv N, Dunham MJ, Kitzman JO, Roth FP, Seelig G, et al. Variant Interpretation: Functional Assays to the Rescue. Am J Hum Genet. 2017;101:315–25.

5. Landrum MJ, Lee JM, Riley GR, Jang W, Rubinstein WS, Church DM, et al. ClinVar: public archive of relationships among sequence variation and human phenotype. Nucleic Acids Res. 2014;42:D980–5.

6. Rollins NJ, Brock KP, Poelwijk FJ, Stiffler MA, Gauthier NP, Sander C, et al. 3D protein structure from genetic epistasis experiments. bioRxiv. 2018;:320721.

7. Schmiedel J, Lehner B. Determining protein structures using genetics. bioRxiv. 2018;:303875.

8. Stiffler MA, Hekstra DR, Ranganathan R. Evolvability as a Function of Purifying Selection in TEM-1 β-Lactamase. Cell. 2015;160:882–92.

9. Lee JM, Huddleston J, Doud MB, Hooper KA, Wu NC, Bedford T, et al. Deep mutational scanning of hemagglutinin helps predict evolutionary fates of human H3N2 influenza variants. Proc Natl Acad Sci. 2018;:201806133.

10. Cantor AJ, Shah NH, Kuriyan J. Deep mutational analysis reveals functional trade-offs in the sequences of EGFR autophosphorylation sites. Proc Natl Acad Sci. 2018;115:E7303–12.

11. Patwardhan RP, Hiatt JB, Witten DM, Kim MJ, Smith RP, May D, et al. Massively parallel functional dissection of mammalian enhancers *in vivo*. Nat Biotechnol. 2012;30:265–70.

12. Chatterjee S, Ahituv N. Gene Regulatory Elements, Major Drivers of Human Disease. Annu Rev Genomics Hum Genet. 2017;18:45–63.

13. Kircher M, Xiong C, Martin B, Schubach M, Inoue F, Bell RJ, et al. Saturation mutagenesis of disease-associated regulatory elements. bioRxiv. 2018;:505362.

14. Gray VE, Hause RJ, Fowler DM. Analysis of Large-Scale Mutagenesis Data To Assess the Impact of Single Amino Acid Substitutions. Genetics. 2017;207:53–61.

15. Gray VE, Hause RJ, Luebeck J, Shendure J, Fowler DM. Quantitative Missense Variant Effect Prediction Using Large-Scale Mutagenesis Data. Cell Syst. 2018;6:116-124.e3.

16. Starita LM, Young DL, Islam M, Kitzman JO, Gullingsrud J, Hause RJ, et al. Massively Parallel Functional Analysis of BRCA1 RING Domain Variants. Genetics. 2015;200:413–22.

17. Majithia AR, Tsuda B, Agostini M, Gnanapradeepan K, Rice R, Peloso G, et al. Prospective functional classification of all possible missense variants in PPARG. Nat Genet. 2016;48:1570–5.

18. Matreyek KA, Starita LM, Stephany JJ, Martin B, Chiasson MA, Gray VE, et al. Multiplex assessment of protein variant abundance by massively parallel sequencing. Nat Genet. 2018;50:874–82.

19. Findlay GM, Daza RM, Martin B, Zhang MD, Leith AP, Gasperini M, et al. Accurate classification of BRCA1 variants with saturation genome editing. Nature. 2018;:1.

20. Starita LM, Islam MM, Banerjee T, Adamovich AI, Gullingsrud J, Fields S, et al. A Multiplex Homology-Directed DNA Repair Assay Reveals the Impact of More Than 1,000 BRCA1 Missense Substitution Variants on Protein Function. Am J Hum Genet. 2018.

21. Chang W, Cheng J, Allaire JJ, Xie Y, McPherson J. shiny: Web Application Framework for R. 2018. https://CRAN.R-project.org/package=shiny.

22. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, et al. NCBI GEO: archive for functional genomics data sets—update. Nucleic Acids Res. 2013;41:D991–5.

23. Leinonen R, Sugawara H, Shumway M. The Sequence Read Archive. Nucleic Acids Res. 2011;39 suppl_1:D19–21.

24. Wang CY, Chang PM, Ary ML, Allen BD, Chica RA, Mayo SL, et al. ProtaBank: A repository for protein design and engineering data. Protein Sci. 2018;27:1113–24.

25. Pires DEV, Blundell TL, Ascher DB. Platinum: a database of experimentally measured effects of mutations on structurally defined protein–ligand complexes. Nucleic Acids Res. 2015;43:D387–91.

26. Bava KA, Gromiha MM, Uedaira H, Kitajima K, Sarai A. ProTherm, version 4.0: thermodynamic database for proteins and mutants. Nucleic Acids Res. 2004;32 suppl_1:D120–1.

27. Bouaoun L, Sonkin D, Ardin M, Hollstein M, Byrnes G, Zavadil J, et al. TP53 Variations in Human Cancers: New Lessons from the IARC TP53 Database and Genomics Data. Hum Mutat. 2016;37:865–76.

28. Gaedigk A, Ingelman-Sundberg M, Miller NA, Leeder JS, Whirl-Carrillo M, Klein TE. The Pharmacogene Variation (PharmVar) Consortium: Incorporation of the Human Cytochrome P450 (CYP) Allele Nomenclature Database. Clin Pharmacol Ther. 2018;103:399–401.

29. Oscarson M, Ingelman-Sundberg M. CYPalleles: A Web Page for Nomenclature of Human Cytochrome P450 Alleles. Drug Metab Pharmacokinet. 2002;17:491–5.

30. Fokkema IFAC, Taschner PEM, Schaafsma GCP, Celli J, Laros JFJ, den Dunnen JT. LOVD v.2.0: the next generation in gene variant databases. Hum Mutat. 2011;32:557–63.

31. The UniProt Consortium. UniProt: the universal protein knowledgebase. Nucleic Acids Res. 2017;45:D158–D169.

32. Fowler DM, Araya CL, Fleishman SJ, Kellogg EH, Stephany JJ, Baker D, et al. High-resolution mapping of protein sequence-function relationships. Nat Methods. 2010;7:741–6.

33. Melamed D, Young DL, Gamble CE, Miller CR, Fields S. Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly(A)-binding protein. RNA. 2013;19:1537–51.

34. Jiang L, Liu P, Bank C, Renzette N, Prachanronarong K, Yilmaz LS, et al. A Balance between Inhibitor Binding and Substrate Processing Confers Influenza Drug Resistance. J Mol Biol. 2016;428:538–53.

35. Plesa C, Sidore AM, Lubock NB, Zhang D, Kosuri S. Multiplexed gene synthesis in emulsions for exploring protein functional landscapes. Science. 2018;359:343–7.

36. Matuszewski S, Hildebrandt ME, Ghenu A-H, Jensen JD, Bank C. A Statistical Guide to the Design of Deep Mutational Scanning Experiments. Genetics. 2016;204:77–87.

37. Rubin AF, Gelman H, Lucas N, Bajjalieh SM, Papenfuss AT, Speed TP, et al. A statistical framework for analyzing deep mutational scanning data. Genome Biol. 2017;18:150.

38. Fowler DM, Araya CL, Gerard W, Fields S. Enrich: software for analysis of protein function by enrichment and depletion of variants. Bioinformatics. 2011;27:3430–1.

39. Hietpas R, Roscoe B, Jiang L, Bolon DNA. Fitness analyses of all possible point mutations for regions of genes in yeast. Nat Protoc. 2012;7:1382–96.

40. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing; 2018. https://www.R-project.org.

41. Creative Commons — Attribution-NonCommercial-ShareAlike 4.0 International — CC BY-NC-SA 4.0. https://creativecommons.org/licenses/by-nc-sa/4.0/.

42. Creative Commons — Attribution 4.0 International — CC BY 4.0. https://creativecommons.org/licenses/by/4.0/.

43. Creative Commons — CC0 1.0 Universal. https://creativecommons.org/publicdomain/zero/1.0/.

44. Weile J, Sun S, Cote AG, Knapp J, Verby M, Mellor JC, et al. A framework for exhaustively mapping functional missense variants. Mol Syst Biol. 2017;13:957.

45. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28:235–42.

46. den Dunnen JT, Dalgleish R, Maglott DR, Hart RK, Greenblatt MS, McGowan-Jordan J, et al. HGVS Recommendations for the Description of Sequence Variants: 2016 Update. Hum Mutat. 2016;37:564–9.

47. Lek M, Karczewski KJ, Minikel EV, Samocha KE, Banks E, Fennell T, et al. Analysis of protein-coding genetic variation in 60,706 humans. Nature. 2016;536:285–91.

48. Yingzhou W, Weile J, Cote A, Sun S, Knapp J, Verby M, et al. A web application and service for imputing and visualizing missense variant effect maps. Bioinformatics. doi:10.1093/bioinformatics/btz012.

49. Python. https://www.python.org/.

50. Django. https://www.djangoproject.com/.

51. PostgreSQL. https://www.postgresql.org/.

52. RabbitMQ. https://www.rabbitmq.com/.

53. Celery. http://www.celeryproject.org/.

54. Saint-Andre P, Klensin J. Uniform Resource Names (URNs). 2017. http://www.rfc-editor.org/info/rfc8141.

55. ORCID. https://orcid.org/.

56. D. Hardt E. The OAuth 2.0 Authorization Framework. 2012. http://www.rfc-editor.org/info/rfc6749.

57. Markdown. https://daringfireball.net/projects/markdown/.

58. Pandoc. https://pandoc.org/.

59. Docker. https://www.docker.com/index.html.

60. Mitternacht S. FreeSASA: An open source C library for solvent accessible surface area calculations. F1000Research. 2016;5:189.

61. Kabsch W, Sander C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983;22:2577–637.

62. Livingstone CD, Barton GJ. Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. Bioinformatics. 1993;9:745–56.

63. Sievers F, Higgins DG. Clustal Omega, Accurate Alignment of Very Large Numbers of Sequences. In: Russell DJ, editor. Multiple Sequence Alignment Methods. Totowa, NJ: Humana Press; 2014. p. 105–16. doi:10.1007/978-1-62703-646-7_6.

64. django-extensions. Python. Django Extensions. https://github.com/django-extensions/django-extensions. Accessed 30 Aug 2018.

65. Graphviz. https://www.graphviz.org/.

Supplemental Figure 1: UML (Unified Markup Language) diagram of the complete MaveDB schema in PDF format. The diagram was generated using the Django Extensions package and visualized using Graphviz [64, 65].

GenomicInterval <TimeStampedModel>

id	**AutoField**
reference_map	**ForeignKey (id)**
chromosome	CharField
creation_date	DateField
end	PositiveIntegerField
modification_date	DateField
start	PositiveIntegerField
strand	CharField

ReferenceMap <TimeStampedModel>

id	**AutoField**
genome	**ForeignKey (id)**
target	**ForeignKey (id)**
creation_date	DateField
is_primary	BooleanField
modification_date	DateField

Variant <UrnModel>

id	**AutoField**
scoreset	**ForeignKey (id)**
creation_date	DateField
data	JSONField
hgvs_nt	TextField
hgvs_pro	TextField
modification_date	DateField
urn	CharField

ScoreSet <DatasetModel>

id	**AutoField**
created_by	ForeignKey (id)
experiment	**ForeignKey (id)**
licence	ForeignKey (id)
modified_by	ForeignKey (id)
replaces	**OneToOneField (id)**
abstract_text	TextField
approved	BooleanField
creation_date	DateField
data_usage_policy	TextField
dataset_columns	JSONField
extra_metadata	JSONField
last_child_value	IntegerField
method_text	TextField
modification_date	DateField
normalised	BooleanField
private	BooleanField
processing_state	CharField
publish_date	DateField
short_description	TextField
title	CharField
urn	CharField

Experiment <DatasetModel>

id	**AutoField**
created_by	ForeignKey (id)
experimentset	ForeignKey (id)
modified_by	**ForeignKey (id)**
abstract_text	TextField
approved	BooleanField
creation_date	DateField
extra_metadata	JSONField
last_child_value	IntegerField
method_text	TextField
modification_date	DateField
private	BooleanField
processing_state	CharField
publish_date	DateField
short_description	TextField
title	CharField
urn	CharField

ExperimentSet <DatasetModel>

id	**AutoField**
created_by	ForeignKey (id)
modified_by	ForeignKey (id)
abstract_text	TextField
approved	BooleanField
creation_date	DateField
extra_metadata	JSONField
last_child_value	IntegerField
method_text	TextField
modification_date	DateField
private	BooleanField
processing_state	CharField
publish_date	DateField
short_description	TextField
title	CharField
urn	CharField

Licence <TimeStampedModel>

id	**AutoField**
creation_date	DateField
legal_code	TextField
link	URLField
long_name	CharField
modification_date	DateField
short_name	CharField
version	CharField

ReferenceGenome <TimeStampedModel>

id	**AutoField**
genome_id	ForeignKey (id)
creation_date	DateField
modification_date	DateField
organism_name	CharField
short_name	CharField

Profile <TimeStampedModel>

id	**AutoField**
user	**OneToOneField (id)**
auth_token	CharField
auth_token_expiry	DateField
creation_date	DateField
email	EmailField
modification_date	DateField

GenomeIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

Keyword <TimeStampedModel>

id	**AutoField**
creation_date	DateField
modification_date	DateField
text	CharField

FailedTask

id	**AutoField**
user	**ForeignKey (id)**
args	TextField
celery_task_id	CharField
creation_date	DateTimeField
exception_class	TextField
exception_msg	TextField
failures	PositiveSmallIntegerField
full_name	TextField
kwargs	TextField
modification_date	DateTimeField
name	CharField
traceback	TextField

DatasetModel <UrnModel>

created_by	**ForeignKey (id)**
modified_by	**ForeignKey (id)**
abstract_text	TextField
approved	BooleanField
creation_date	DateField
extra_metadata	JSONField
last_child_value	IntegerField
method_text	TextField
modification_date	DateField
private	BooleanField
processing_state	CharField
publish_date	DateField
reference_html	TextField
short_description	TextField
title	CharField

PubmedIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
reference_html	TextField
url	URLField

SraIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

DoiIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

PublicDatasetCounter <TimeStampedModel>

id	**AutoField**
creation_date	DateField
experimentsets	IntegerField
modification_date	DateField

UrnModel <TimeStampedModel>

creation_date	DateField
modification_date	DateField

User

TimeStampedModel

creation_date	DateField
modification_date	DateField

EnsemblOffset <AnnotationOffset>

id	**AutoField**
identifier	**ForeignKey (id)**
target	ForeignKey (id)
offset	PositiveIntegerField

UniprotOffset <AnnotationOffset>

id	**AutoField**
identifier	**ForeignKey (id)**
target	ForeignKey (id)
offset	PositiveIntegerField

RefseqOffset <AnnotationOffset>

id	**AutoField**
identifier	**ForeignKey (id)**
target	ForeignKey (id)
offset	PositiveIntegerField

AnnotationOffset

target	**ForeignKey (id)**
offset	PositiveIntegerField

TargetGene <TimeStampedModel>

id	**AutoField**
ensembl_id	ForeignKey (id)
refseq_id	ForeignKey (id)
scoreset	**OneToOneField (id)**
uniprot_id	ForeignKey (id)
wt_sequence	**OneToOneField (id)**
category	CharField
creation_date	DateField
modification_date	DateField
name	CharField

EnsemblIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

UniprotIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

RefseqIdentifier <ExternalIdentifier>

id	**AutoField**
creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

WildTypeSequence <TimeStampedModel>

id	**AutoField**
creation_date	DateField
modification_date	DateField
sequence	TextField

ExternalIdentifier <TimeStampedModel>

creation_date	DateField
dbname	CharField
dbversion	CharField
identifier	CharField
modification_date	DateField
url	URLField

News <TimeStampedModel>

id	**AutoField**
creation_date	DateField
level	CharField
modification_date	DateField
text	TextField

SiteInformation <TimeStampedModel>

id	**AutoField**
branch	CharField
creation_date	DateField
email	EmailField
md_about	TextField
md_citation	TextField
md_documentation	TextField
md_privacy	TextField
md_terms	TextField
md_usage_guide	TextField
modification_date	DateField
version	CharField
version_date	DateField

reference_map (intervals)
target (reference_maps)
genome (associated_reference_maps)
scoreset (variants)
licence (attached_scoresets)
experiment (scoresets)
replaces (replaced_by)
genome_id (associated_referencegenomes)
keywords (associated_scoresets)
keywords (associated_experiments)
keywords (associated_experimentsets)
experimentset (experiments)
pubmed_ids (associated_scoresets)
pubmed_ids (associated_experiments)
pubmed_ids (associated_experimentsets)
sra_ids (associated_scoresets)
sra_ids (associated_experiments)
sra_ids (associated_experimentsets)
doi_ids (associated_scoresets)
doi_ids (associated_experiments)
doi_ids (associated_experimentsets)
identifier (offset)
target (%(class)s)
scoreset (target)
ensembl_id (associated_targetgenes)
uniprot_id (associated_targetgenes)
wt_sequence (target)
refseq_id (associated_targetgenes)
user (profile)
user (failed_tasks)
created_by (last_created_%(class)s)
modified_by (last_edited_%(class)s)
abstract inheritance