pVACtools: a computational toolkit to select and visualize cancer neoantigens

Jasreet Hundal[1+], Susanna Kiwala[1+], Joshua McMichael[1], Christopher A. Miller[1,2,4], Alexander T. Wollam[1], Huiming Xia[1], Connor J. Liu[1], Sidi Zhao[1] ,Yang-Yang Feng[1], Aaron P. Graubert[1], Amber Z. Wollam[1], Jonas Neichin[1], Megan Neveau[1], Jason Walker[1], William E Gillanders[4,5], Elaine R. Mardis[3], Obi L. Griffith[1,2,4,6,*], Malachi Griffith[1,2,4,6,*]

Affiliations

(1) McDonnell Genome Institute, Washington University School of Medicine, St. Louis, MO, USA
(2) Division of Oncology, Department of Medicine, Washington University School of Medicine, St. Louis, MO, USA
(3) Institute for Genomic Medicine, Nationwide Children's Hospital, 575 Children's Crossroad, Columbus OH, USA
(4) Siteman Cancer Center, Washington University School of Medicine, St. Louis, MO, USA
(5) Department of Surgery, Washington University School of Medicine, St. Louis, MO, USA
(6) Department of Genetics, Washington University School of Medicine, St. Louis, MO, USA

+ These authors contributed equally to this work
* Corresponding authors. obigriffith@wustl.edu, mgriffit@wustl.edu

Abstract

The elucidation of neoantigens is a critical step in predicting response to checkpoint blockade therapy and design of personalized cancer vaccines. We have developed an *in silico* sequence analysis method - pVACtools, to facilitate comprehensive neoantigen characterization. This modular workflow consists of tools for neoantigen prediction from somatic alterations (pVACseq and pVACfuse), prioritization and selection using a graphical web-based interface (pVACviz), and determining the optimal order of neoantigen candidates in a DNA vector-based vaccine(pVACvector).

Increasing interest in identifying the numbers and types of predicted neoantigens encoded by a cancer genome has placed an emphasis on the facility and precision of related computational prediction tools[1]. Several such efforts have been published[2,3,4]. Typically, these tools start with a list of somatic variants (in VCF or other formats), with annotated protein changes, and predict the strongest MHC binding peptides (8-11-mer for class I MHC and 13-25-mer for class II) using one or more prediction algorithms[5,6,7]. The predicted neoantigens are then filtered or ranked based on defined metrics including sequencing read coverage, variant allele fraction, gene expression, and differential binding compared to the wild type peptide (agretopicity index score[8]). However, of the small number of such prediction tools **(Supp Table 1)**, most lack some key functionality, including predicting neoantigens from gene fusions, aiding optimized vaccine design for DNA cassette vaccines, and including nearby germline or somatic alterations into the candidate neoantigens[9]. An intuitive graphical user interface to visualize and efficiently select the most promising candidates is also critical to facilitate involvement of clinicians and other researchers in the process of neoantigen evaluation.

To address these limitations and to add facility for all end-users, we created a comprehensive and extensible framework for computational identification, selection, prioritization and visualization of neoantigens - *'pVACtools',* that facilitates each of the major components of neoantigen identification. This computational framework can be used to identify neoantigens from a variety of somatic alterations, including gene fusions and insertion/deletion frameshift mutations, both of which potentially create very strong immunogenic neoantigens[10]. Further, pVACtools can facilitate both MHC class I and II predictions, and provides an interactive display of predicted neoantigens for review by the end user.

The pVACtools workflow (**Fig 1**) is divided into flexible components that can be run independently. The main tools in the workflow are: (a) pVACseq: a significantly enhanced and reengineered version of our previous pipeline[11] for identifying and prioritizing neoantigens from a variety of tumor-specific alterations (b) pVACfuse: a tool for detecting neoantigens resulting from gene fusions (c) pVACviz: a graphical user interface web client for process management, visualization and selection of results from pVACseq (d) pVACvector: a tool for optimizing design of neoantigens and nucleotide spacers in a DNA vector that prevents high-affinity junctional epitopes, and (e) pVACapi: an OpenAPI HTTP REST interface to the pVACtools suite.

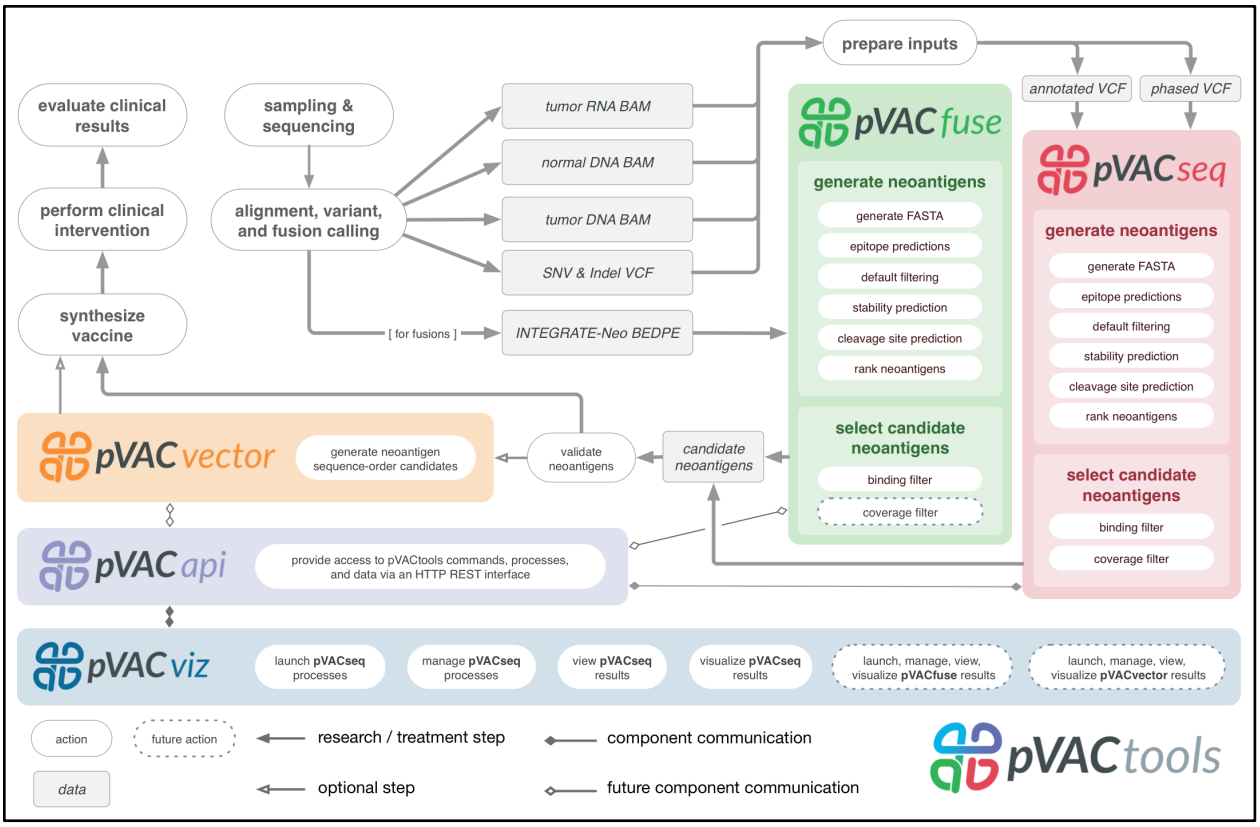


Figure 1: Overview of pVACtools workflow: *The pVACtools workflow is highly modularized and is divided into flexible components that can be run independently. The main tools under the workflow include pVACseq[11] for identifying and prioritizing neoantigens from a variety of somatic alterations (red inset box), pVACfuse (green) for detecting neoantigens resulting from gene fusions, pVACviz (blue) for process management, visualization and selection of results and pVACvector (orange) for optimizing design of neoantigens and nucleotide spacers in a DNA vector. All of these tools interact via the pVACapi (purple), an OpenAPI HTTP REST interface to the pVACtools suite.*

pVACseq[11] has been completely implemented in Python3 and extended to include many new features since our initial report of its use. pVACseq no longer requires a custom input format for variants, and now uses a standard VCF file annotated with VEP[12]. In our own neoantigen identification pipeline, this VCF is the result of merging results from multiple somatic variant callers and RNA expression tools (**Methods**). Information that is not natively available in the VCF output from somatic variant callers (such as coverage and variant allele fractions for RNA and DNA, as well as gene and transcript expression values) now can be added to the VCF using vcf-annotation-tools (vatools.org), a suite of accessory scripts that we created to accompany pVACtools. pVACtools queries these features directly from the VCF, enabling prioritization and filtering of neoantigen candidates based on sequence coverage and expression information. In addition, pVACtools now makes use of phasing information provided in the VCF, taking into account all variants proximal to somatic variants of interest that alter neoantigen sequences[9]. Since proximal variants can change the neoantigenic

peptide sequence and also affect neoantigen binding predictions, this is an important confounding factor to ensure that the selected neoantigens correctly represent the individual's genome. We have also expanded the supported mutation types for neoantigen predictions to include in-frame indels and frameshift mutations. These capabilities expand the potential number of targetable neoantigens several-fold in many tumors (**Supplementary Data**).

To prioritize neoantigens, pVACseq now offers support for as many as eight different MHC Class I epitope prediction algorithms and four MHC Class II prediction algorithms. The tool does this by leveraging the Immune Epitope Database (IEDB)[13] and their suite of six different MHC class I prediction algorithms, as well as three MHC Class II algorithms (**Methods**). pVACseq supports local installation of these tools for power-users, or provides straightforward access by default via the IEDB RESTful web interface. In addition, pVACseq now contains an extensible framework for supporting new neoantigen prediction algorithms that has been used to add support for two new non-IEDB algorithms - MHCflurry[14] and MHCnuggets[15]. By creating a framework that integrates many tools we allow for (a) a broader ensemble approach than IEDB, and (b) a system that other users can leverage to develop improved ensemble ranking, or to integrate proprietary or not-yet-public prediction software. Importantly, this framework enables non-informatics-savvy users to predict neoantigens from sequence variant data sets.

Once neoantigens have been predicted, the pVACseq ranking score is used to prioritize them. This score takes into account gene expression, sequence read coverage, binding affinity predictions, and agretopicity (**Methods**). In addition to applying strict binding affinity cutoffs, the pipeline also offers support for MHC allele-specific cutoffs[16]. Taking a step further than most commonly used approaches, we also offer cleavage position predictions via optional processing through NetChop[17] as well as stability predictions made by NetMHCstab[18].

Previous studies have shown that the novel protein sequences produced by gene fusions frequently produce neoantigen candidates[19]. pVACfuse provides support for predicting neoantigens from such gene fusions. Fusion variants may be imported in annotated BEDPE format from any fusion caller (we used INTEGRATE-Neo[19]). These variants are then assessed for presence of fusion neo-epitopes using predictions against any of the pVACseq-supported binding prediction algorithms.

Implementing cancer vaccines in a clinical setting requires multidisciplinary teams, many of whom may not be informatics savvy. To support this growing community of users, we developed pVACviz, which is a browser-based user interface that assists in

launching, managing, reviewing, and visualizing the results of pVACtools processes. Instead of interacting with the tools via terminal/shell commands, the pVACviz client provides a modern web-based user experience. Users complete a pVACseq process setup form that provides helpful documentation and suggests valid values for inputs. The client also provides views showing ongoing processes, their logs, and interim data files to aid in managing and troubleshooting. After a process has completed, users may examine the results as a filtered data table, or as a scatterplot visualization - allowing them to curate results and save them as a CSV file for further analysis.

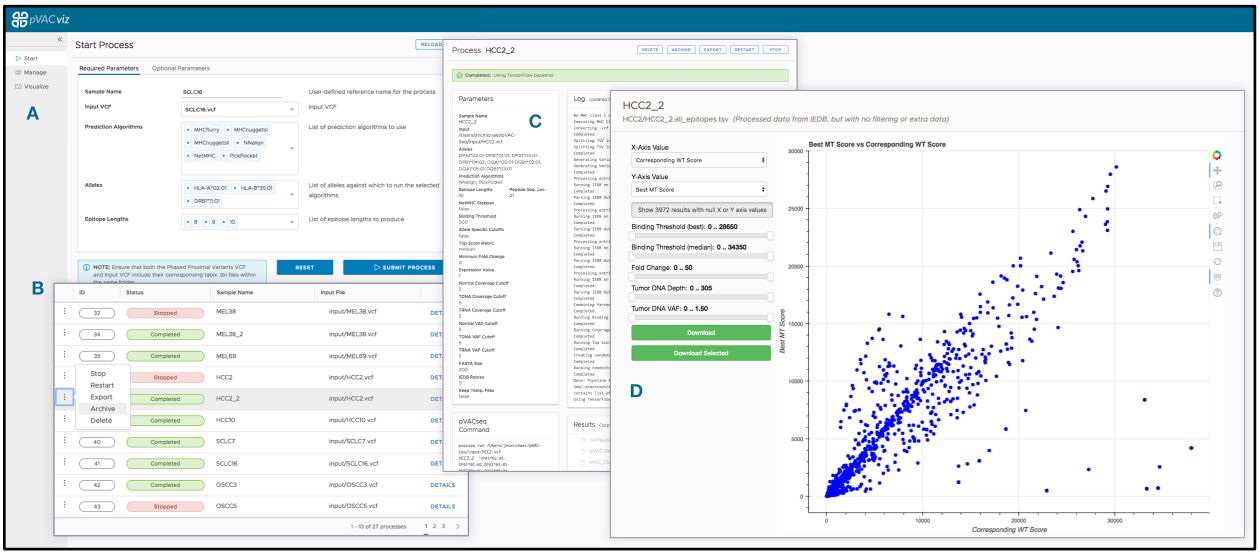


Figure 2: pVACviz GUI client: *pVACtools provides a browser-based graphic user interface, called pVACviz, that provides an intuitive means to launch pipeline processes, monitor their execution, and analyze, export, or archive their results. To launch a process, users navigate to the Start Page (A), and complete a form containing all of the relevant inputs and settings for a pVACseq process. Each form field includes help text, and provides typeahead completion where applicable. For instance, the Alleles field provides a typeahead dropdown menu that match available alleles. Once a process is launched, a user may monitor its progress on the Manage Page (B), which lists all running, stopped, and completed processes. The Details Page (C) shows a process' current log, attributes, and any results files as well as providing buttons for stopping, restarting, exporting and archiving the process. The results of pipeline processes may be analyzed on the Visualize Page (D), which displays a customizable scatterplot of a file's rows. The X and Y axis may be set to any column in the result set, and filters may be applied to values in any column. Additionally, points may be selected on the scatter plot or data grid (not visible in this figure) for further analysis or export as CSV files.*

Furthermore, to support informatics groups that want to incorporate or build upon the pVACtools features, we developed pVACapi, which provides a HTTP REST interface to the pVACtools suite. Currently, it provides the API that pVACviz uses to interact with the pVACtools suite. Advanced users could develop their own user interfaces, or use the API to control multiple pVACtools installations remotely over an HTTP network.

Once a list of neoantigen candidates has been prioritized and selected, the pVACvector utility can be used to aid in the construction of DNA-based cancer vaccines. The input is either the output file from pVACseq or a fasta file containing peptide sequences, and pVACvector returns a neoantigen sequence ordering that minimizes the effects of junctional epitopes (which may create novel antigens) between the sequences. This is accomplished by using the core pVACseq services to predict the binding scores for each junctional peptide and by testing junctions with spacer[20] amino acid sequences that may help to reduce reactivity. The final vaccine ordering is achieved through a simulated annealing procedure that returns a near-optimal solution, when one exists.

pVACtools has been used to predict and prioritize neoepitopes for several neoantigen studies[21–23] and cancer vaccine clinical trials (e.g. NCT02348320 and NCT03122106). We also have a large external user community (the original 'pvacseq' package has been downloaded over 37,000 times from PyPi, and the 'pvactools' package has been downloaded over 9,000 times) that has been actively evaluating and using these packages for their neoantigen analysis, and has also helped in the subsequent refinement of pVACtools through feedback.

To demonstrate the utility and performance of the pVACtools package, we downloaded exome sequencing and RNA-Seq data from The Cancer Genome Atlas (TCGA)[24] from 100 cases each of melanoma, hepatocellular carcinoma and lung squamous cell carcinoma, and used patient-specific MHC Class I alleles (**Supp Fig 1**) to determine neoantigen candidates for each cancer. By extending support for additional variant types as well as prediction algorithms, we produced 42% more neoantigens versus the previous version of pVACseq[11]. **(Supplementary Data)**

As reported from our demonstration analysis, a typical tumor has too many possible neoantigen candidates to be practical for a vaccine. There is therefore a critical need for a tool that takes in the input from a standard sequencing analysis pipeline and reports a filtered and prioritized list of neoantigens. pVACtools enables a streamlined, accurate and user-friendly analysis of neoantigenic peptides from NGS cancer datasets. This suite offers a complete and easily configurable end-to-end analysis, starting from somatic variants and gene fusions (pVACseq and pVACfuse respectively), through filtering, prioritization, and visualization of candidates (pVACviz), and determining the best arrangement of candidates for a DNA vector vaccine (pVACvector). Furthermore, by supporting additional classes of variants as well as gene fusions, we offer an increase in the number of predicted epitopes which is even more important in the case of low mutational burden tumors. Finally, by extending support for multiple binding prediction algorithms, we allow for a consensus approach. The need for this integrated

approach is made abundantly clear by the high disagreement between these algorithms observed in our demonstration analyses (**Supplementary Data**).

The results from pVACtools analyses are already being used in cancer immunology studies, including studying the relationship between tumor mutation burden and neoantigen load to predict response in checkpoint blockade therapy trials and the design of cancer vaccines in ongoing clinical trials. We anticipate that pVACtools will make such analyses more robust, reproducible, and facile as these efforts continue.

Additional information

Online Methods:

TCGA data pre-processing

Aligned tumor and normal BAMs from BWA[25] (version 0.7.12-r1039) as well as somatic variant calls from VarScan2[26,27] (in VCF format) were downloaded from the Genomic Data Commons (GDC, https://gdc.cancer.gov/). Since the GDC does not provide germline variant calls for TCGA data, we used GATK's[28] HaplotypeCaller to perform germline variant calling using default parameters. These calls were refined using VariantRecalibrator in accordance with GATK Best Practices[29]. Somatic and germline missense variant calls from each sample were then combined using GATK's CombineVariants, and the variants were subsequently phased using GATK's ReadBackedPhasing algorithm.

Phased Somatic VCF files were annotated with RNA depth and expression information using VCF annotation tools (vatools.org). We restricted our analysis to only consider 'PASS' variants in these VCFs as these are higher confidence than the raw set, and the variants were annotated using the "--pick" option in VEP.

Existing *in silico* HLA typing information was obtained from The Cancer Immunome Atlas (TCIA) database[30].

Neoantigen prediction

The VEP-annotated VCF files were then run through pVACseq using all eight Class I prediction algorithms and for epitope lengths 8-11. The current MHC Class I algorithms supported by pVACseq are NetMHCpan[31], NetMHC[6,31], NetMHCcons[32], PickPocket[33], SMM[34], SMMPMBEC[35], MHCflurry[14] and MHCnuggets[15]. The four MHC Class II algorithms that are supported are NetMHCIIpan, SMMalign, NNalign, and MHCnuggets. For the demonstration analysis, we limited our prediction to only MHC Class I alleles due to availability of HLA typing information from TCIA, though predictions of Class II can be just as easily generated using pVACtools.

Ranking of Neoantigens
To help prioritize neoantigens, a ranking score is assigned where each of the following four criteria are assigned a rank-ordered value (where the worst = 1):

B = binding affinity
F = Fold Change between MT and WT alleles
M = mutant allele expression, calculated as (Gene expression * Mutant allele RNA Variant allele fraction)
D = DNA Variant allele fraction

A final ranking is based on a score obtained from combining these values:
Priority Score = B+F+(M*2)+(D/2). This score is not meant to be the final word on peptide suitability for vaccines, but was designed to be a useful metric.

Pipeline for creation of pVACtools input files
pVACtools is designed to support a standard VCF variant file format and thus, should be compatible with many existing variant calling pipelines. However, as a reference, we provide the following description of our current somatic and expression analysis pipeline (manuscript in preparation) which has been implemented using docker, CWL[36], and Cromwell[37]. The pipeline consists of workflows for alignment of exome/DNA- and RNA-Seq data, somatic and germline variant detection, RNA-Seq expression estimation as well as optional HLA typing.

This pipeline starts with raw patient tumor exome or cDNA capture[38] and RNA-seq data and produces annotated VCFs for neoantigen identification and prioritization with pVACtools. Our pipeline consists of three main components: DNA alignment, variant detection and annotation, as well as RNA-seq data processing. More specifically, we use BWA-MEM[25] for aligning the patient's tumor and normal exome data. The output BAM then undergoes merging (Samtools[39] Merge), query name sort (Picard SortSam), duplicate marking (Picard MarkDuplicates), position sorting followed by base quality recalibration (GATK BaseRecalibrator). GATK's HaplotypeCaller[28] is used for germline variant calling and the output variants are annotated using VEP[12] and filtered for coding sequence variants.

For somatic variant calling, our pipeline combines the output of four variant detection algorithms- Mutect2[40], Strelka[41], Varscan[26,27] and Pindel[42]. The combined variants are normalized using GATK's LeftAlignAndTrimVariants where the indels are left-aligned and common bases are trimmed. Vt[43] is used to split multi-allelic variants. Several filters such as gnomAD allele frequency, percentage of mapq0 reads, as well as pass-only variants are applied prior to annotation of the VCF using VEP. We use a combination of

custom and standard plugins for VEP annotation (parameters: --format VCF --plugin Downstream --plugin Wildtype --symbol --term SO --transcript_version --tsl --coding_only --flag_pick --hgvs). Variant coverage is assessed using bam-readcount (https://github.com/genome/bam-readcount) for both the tumor and normal DNA exome data and is also annotated into the VCF output using VCF-annotation-tools(vatools.org).

Our pipeline also generates a phased-VCF file by combining both the somatic and germline variants and running the sorted combined variants through GATK ReadBackedPhasing.

For RNA-seq data, the pipeline first trims the adapter sequence using flexbar[44] and aligns the patient's tumor RNA-seq data using HISAT2[45]. Two different methods, Stringtie[46] and Kallisto[47], are employed for evaluating both the transcript and gene expression values. Additionally, coverage support for variants in RNA-seq data can also be assessed through bam-readcount. This information is added to the VCF using VCF-annotation-tools and serves as an input for neoantigen prioritization using pVACtools.

Optionally, our pipeline can also run HLA-typing *in silico* using OptiType[48] when clinical HLA typing is not available.

Implementation

pVACtools is written in Python3. The individual tools are implemented as separate command line entry points that can be run using the `pvacseq`, `pvacfuse`, `pvacvector`, `pvacapi`, and `pvacviz` commands to run the respective tool. pVACapi is required to run pVACviz so both the `pvacapi` and `pvacviz` command need to be executed in separate terminals. For pVACseq, the PyVCF package is used for parsing the input VCF files. The `mhcflurry` and `mhcnuggets` packages are used to run the MHCflurry and MHCnuggets prediction algorithms, respectively. The `pandas` package is used for data management while filtering and ranking the neoantigen candidates in pVACseq and pVACfuse. The `simanneal` package is used for the simulated annealing procedure when running pVACvector. pVACapi is implemented using Flask and Bokeh. The pVACviz client is written in TypeScript using the Angular web application framework, the Clarity UI component library, and the ngrx library for managing application state. The test suite is implemented using the Python unittest framework and GitHub integration tests are run using travis-ci (travis-ci.org). Code changes are integrated using GitHub pull requests (https://github.com/griffithlab/pVACtools/pulls). Feature additions, user requests, and bug reports are managed using the GitHub issue tracking (https://github.com/griffithlab/pVACtools/issues). User documentation is written using

the reStructuredText markup language and the Sphinx documentation framework (sphinx-doc.org). Documentation is hosted on Read The Docs (readthedocs.org).

Data availability

Data from 100 cases each of melanoma, hepatocellular carcinoma and lung squamous cell carcinoma were obtained from TCGA and downloaded via the Genomics Data Commons (GDC). This data can be accessed under dbGaP study accession phs000178. Data for demonstration and analysis of fusion neoantigens was downloaded from the Github repo for Integrate (https://github.com/ChrisMaherLab/INTEGRATE-Vis/tree/master/example).

Software availability:

The pVACtools codebase is hosted publicly on GitHub at https://github.com/griffithlab/pVACtools and https://github.com/griffithlab/BGA-interface-projects (pVACviz). User documentation is available at pvactools.org. This project is licensed under the Non-Profit Open Software License version 3.0 (NPOSL-3.0, https://opensource.org/licenses/NPOSL-3.0). pVACtools has been packaged and uploaded to PyPi under the “pvactools” package name and can be installed on Linux systems by running the `pip install pvactools[API]` command. Installation requires a Python 3.5 environment which can be emulated by using Conda. Versioned Docker images are available on DockerHub (https://hub.docker.com/r/griffithlab/pvactools/).

Acknowledgements

We thank the patients and their families for donation of their samples and participation in clinical trials. We also thank our growing user community for testing the software and providing useful input, critical bug reports as well as suggestions for improvement and new features. We are grateful to Drs. Robert Schreiber, Gavin Dunn and Beatriz Carreno for their expertise and guidance on foundational work on cancer immunology using neoantigens and suggestions on improving the pipeline.

Author contributions

JH and SK were involved in all aspects of this study including designing and developing the methodology, and writing the manuscript, with help from CAM, ERM, OLG and MG. JH analyzed and interpreted data with input from HX, CJL, SZ and Y-YF. SK wrote the software code with help from ATW, APG, AZW, MN and JW. JM developed pVACviz and pVACapi with assistance from APG. JN, MN, and CAM developed pVACvector. WEG provided input about clinical needs for vaccine design to help improve the tool features. OG and MG supervised the project and revised the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

References

1. Liu, X. S., Shirley Liu, X. & Mardis, E. R. Applications of Immunogenomics to Cancer. *Cell* **168,** 600–612 (2017).
2. Bjerregaard, A.-M., Nielsen, M., Hadrup, S. R., Szallasi, Z. & Eklund, A. C. MuPeXI: prediction of neo-epitopes from tumor sequencing data. *Cancer Immunol. Immunother.* (2017). doi:10.1007/s00262-017-2001-3
3. Rubinsteyn, A., Hodes, I., Kodysh, J. & Hammerbacher, J. Vaxrank: A Computational Tool For Designing Personalized Cancer Vaccines. (2017). doi:10.1101/142919
4. Bais, P., Namburi, S., Gatti, D. M., Zhang, X. & Chuang, J. H. CloudNeo: a cloud pipeline for identifying patient-specific tumor neoantigens. *Bioinformatics* **33,** 3110–3112 (2017).
5. Andreatta, M. & Nielsen, M. Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics* **32,** 511–517 (2016).
6. Nielsen, M. *et al.* Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Sci.* **12,** 1007–1017 (2003).
7. Jurtz, V. *et al.* NetMHCpan-4.0: Improved Peptide–MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *The Journal of Immunology* **199,** 3360–3368 (2017).
8. Duan, F. *et al.* Genomic and bioinformatic profiling of mutational neoepitopes reveals new rules to predict anticancer immunogenicity. *J. Exp. Med.* **211,** 2231–2248 (2014).
9. Hundal, J. *et al.* Accounting for proximal variants improves neoantigen prediction. *Nat. Genet.* (2018). doi:10.1038/s41588-018-0283-9
10. Turajlic, S. *et al.* Insertion-and-deletion-derived tumour-specific neoantigens and the immunogenic phenotype: a pan-cancer analysis. *Lancet Oncol.* **18,** 1009–1021 (2017).

11. Hundal, J. *et al.* pVAC-Seq: A genome-guided in silico approach to identifying tumor neoantigens. *Genome Med.* **8,** 11 (2016).
12. McLaren, W. *et al.* The Ensembl Variant Effect Predictor. (2016). doi:10.1101/042374
13. Vita, R. *et al.* The Immune Epitope Database 2.0. *Nucleic Acids Res.* **38,** D854–D862 (2009).
14. O'Donnell, T. J. *et al.* MHCflurry: Open-Source Class I MHC Binding Affinity Prediction. *Cell Syst* **7,** 129–132.e4 (2018).
15. Bhattacharya, R. *et al.* Evaluation of machine learning methods to predict peptide binding to MHC Class I proteins. (2017). doi:10.1101/154757
16. Paul, S. *et al.* HLA Class I Alleles Are Associated with Peptide-Binding Repertoires of Different Size, Affinity, and Immunogenicity. *The Journal of Immunology* **191,** 5831–5839 (2013).
17. Keşmir, C., Nussbaum, A. K., Schild, H., Detours, V. & Brunak, S. Prediction of proteasome cleavage motifs by neural networks. *Protein Eng.* **15,** 287–296 (2002).
18. Jørgensen, K. W., Rasmussen, M., Buus, S. & Nielsen, M. NetMHCstab - predicting stability of peptide-MHC-I complexes; impacts for cytotoxic T lymphocyte epitope discovery. *Immunology* **141,** 18–26 (2014).
19. Zhang, J., Mardis, E. R. & Maher, C. A. INTEGRATE-neo: a pipeline for personalized gene fusion neoantigen discovery. *Bioinformatics* **33,** 555–557 (2017).
20. Schubert, B. & Kohlbacher, O. Designing string-of-beads vaccines with optimal spacers. *Genome Med.* **8,** 9 (2016).
21. Miller, A. *et al.* High somatic mutation and neoantigen burden are correlated with decreased progression-free survival in multiple myeloma. *Blood Cancer J.* **7,** e612 (2017).
22. Balachandran, V. P. *et al.* Identification of unique neoantigen qualities in long-term survivors of pancreatic cancer. *Nature* **551,** 512 (2017).
23. Formenti, S. C. *et al.* Radiotherapy induces responses of lung cancer to CTLA-4 blockade.

Nat. Med. **24,** 1845 (2018).

24. Cancer Genome Atlas Research Network *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45,** 1113–1120 (2013).
25. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25,** 1754–1760 (2009).
26. Koboldt, D. C., Larson, D. E. & Wilson, R. K. Using VarScan 2 for Germline Variant Calling and Somatic Mutation Detection. in *Current Protocols in Bioinformatics* 15.4.1–15.4.17 (2013).
27. Koboldt, D. C. *et al.* VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.* **22,** 568–576 (2012).
28. DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43,** 491–498 (2011).
29. Van der Auwera, G. A. *et al.* From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. *Curr. Protoc. Bioinformatics* **43,** 11.10.1–33 (2013).
30. Charoentong, P. *et al.* Pan-cancer Immunogenomic Analyses Reveal Genotype-Immunophenotype Relationships and Predictors of Response to Checkpoint Blockade. *Cell Rep.* **18,** 248–262 (2017).
31. Hoof, I. *et al.* NetMHCpan, a method for MHC class I binding prediction beyond humans. *Immunogenetics* **61,** 1–13 (2009).
32. Karosiene, E., Lundegaard, C., Lund, O. & Nielsen, M. NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. *Immunogenetics* **64,** 177–186 (2011).
33. Zhang, H., Lund, O. & Nielsen, M. The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: application to MHC-peptide binding. *Bioinformatics* **25,** 1293–1299 (2009).
34. Peters, B. & Sette, A. 10.1186/1471-2105-6-132. *BMC Bioinformatics* **6,** 132 (2005).

35. Kim, Y., Sidney, J., Pinilla, C., Sette, A. & Peters, B. Derivation of an amino acid similarity matrix for peptide: MHC binding and its application as a Bayesian prior. *BMC Bioinformatics* **10,** 394 (2009).

36. Peter Amstutz, Michael R. Crusoe, Nebojša Tijanić (editors), Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, Matt Scales, Stian Soiland-Reyes, Luka Stojanovic. Common Workflow Language, v1.0. (2016). doi:10.6084/m9.figshare.3115156.v2

37. Voss K, G. J. A. V. der A. G. Full-stack genomics pipelining with GATK4 + WDL + Cromwell [version 1; not peer reviewed]. *F1000Res.* (2017). doi:10.7490/f1000research.1114631.1

38. Cabanski, C. R. *et al.* cDNA hybrid capture improves transcriptome analysis on low-input and archived samples. *J. Mol. Diagn.* **16,** 440–451 (2014).

39. Li H, E. al. The Sequence Alignment/Map format and SAMtools. - PubMed - NCBI. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19505943. (Accessed: 19th December 2018)

40. Cibulskis, K. *et al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.* **31,** 213–219 (2013).

41. Saunders, C. T. *et al.* Strelka: accurate somatic small-variant calling from sequenced tumor-normal sample pairs. *Bioinformatics* **28,** 1811–1817 (2012).

42. Ye, K., Schulz, M. H., Long, Q., Apweiler, R. & Ning, Z. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* **25,** 2865–2871 (2009).

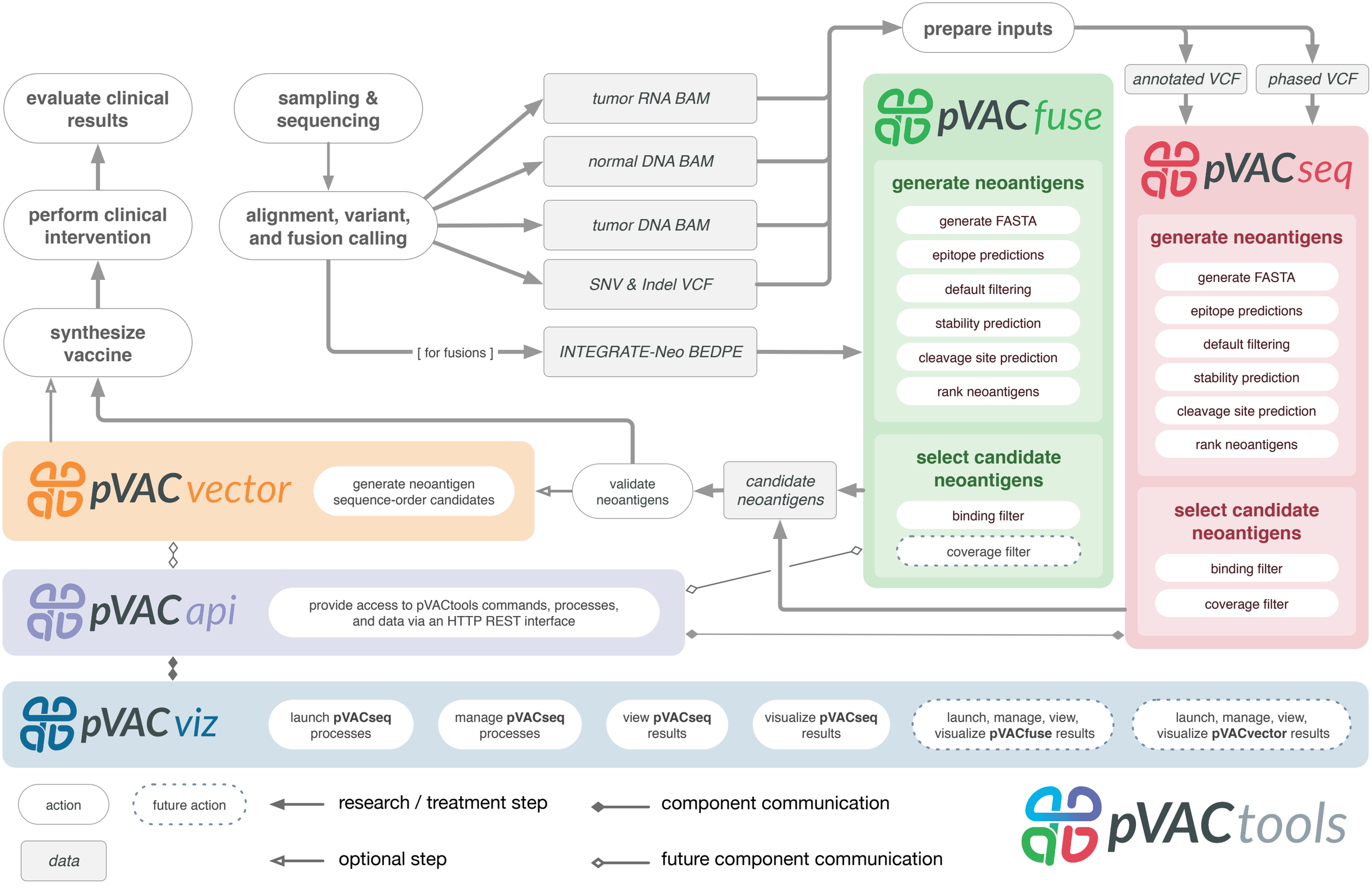
43. Tan, A., Abecasis, G. R. & Kang, H. M. Unified representation of genetic variants. *Bioinformatics* **31,** 2202–2204 (2015).

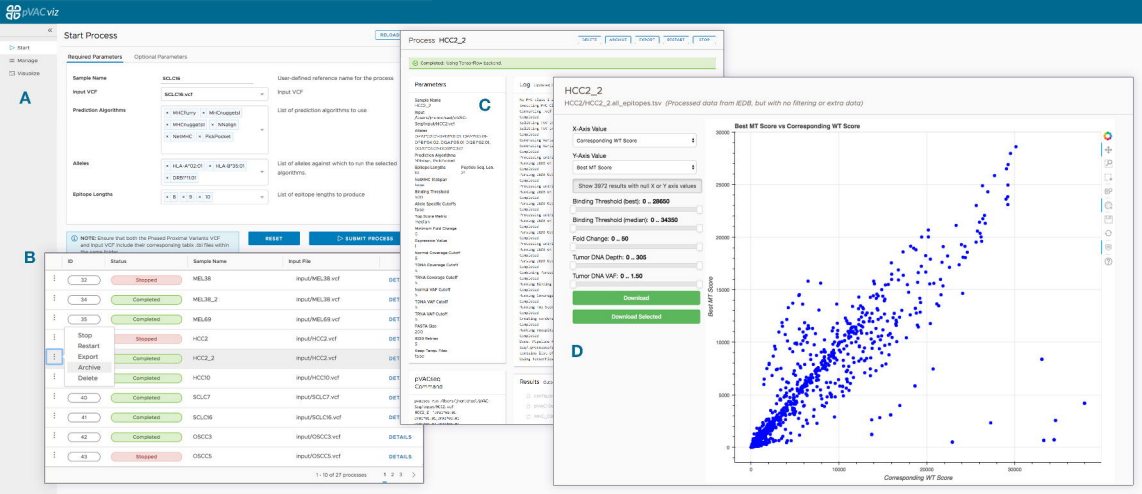
44. Roehr, J. T., Dieterich, C. & Reinert, K. Flexbar 3.0 - SIMD and multicore parallelization. *Bioinformatics* **33,** 2941–2942 (2017).

45. Kim, D., Langmead, B. & Salzberg, S. L. HISAT: a fast spliced aligner with low memory

requirements. *Nat. Methods* **12,** 357–360 (2015).

46. Pertea, M. *et al.* StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat. Biotechnol.* **33,** 290–295 (2015).
47. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34,** 525–527 (2016).
48. Szolek, A. *et al.* OptiType: precision HLA typing from next-generation sequencing data. *Bioinformatics* **30,** 3310–3316 (2014).


evaluate clinical results
perform clinical intervention
synthesize vaccine
sampling & sequencing
alignment, variant, and fusion calling
tumor RNA BAM
normal DNA BAM
tumor DNA BAM
SNV & Indel VCF
[for fusions]
INTEGRATE-Neo BEDPE
prepare inputs
annotated VCF
phased VCF
pVACfuse
generate neoantigens
generate FASTA
epitope predictions
default filtering
stability prediction
cleavage site prediction
rank neoantigens
select candidate neoantigens
binding filter
coverage filter
pVACseq
generate neoantigens
generate FASTA
epitope predictions
default filtering
stability prediction
cleavage site prediction
rank neoantigens
select candidate neoantigens
binding filter
coverage filter
candidate neoantigens
validate neoantigens
pVACvector
generate neoantigen sequence-order candidates
pVACapi
provide access to pVACtools commands, processes, and data via an HTTP REST interface
pVACviz
launch pVACseq processes
manage pVACseq processes
view pVACseq results
visualize pVACseq results
launch, manage, view, visualize pVACfuse results
launch, manage, view, visualize pVACvector results
action
future action
data
research / treatment step
optional step
component communication
future component communication
pVACtools


pVAC viz
Start
Manage
Visualize
A
Start Process
Required Parameters
Optional Parameters
Sample Name
SCLC16
User-defined reference name for the process
Input VCF
SCLC16.vcf
Input VCF
Prediction Algorithms
MHCflurry
MHCnuggetsI
MHCnuggetsII
NetMHC
PickPocket
List of prediction algorithms to use
Alleles
HLA-A*02:01
HLA-B*35:01
DRB1*11:01
List of alleles against which to run the selected algorithms.
Epitope Lengths
8
9
10
List of epitope lengths to produce
NOTE: Ensure that both the Phased Proximal Variants VCF and Input VCF include their corresponding tabix. Bit files within
RESET
SUBMIT PROCESS
B
ID
Status
Sample Name
Input File
32
Stopped
MEL38
input/MEL38.vcf
34
Completed
MEL38_2
input/MEL38.vcf
35
Completed
MEL69
input/MEL69.vcf
Stop
Restart
Export
Archive
Delete
Stopped
HCC2
input/HCC2.vcf
Completed
HCC2_2
input/HCC2.vcf
Completed
HCC10
input/HCC10.vcf
40
Completed
SCLC7
input/SCLC7.vcf
41
Completed
SCLC16
input/SCLC16.vcf
42
Completed
OSCC3
input/OSCC3.vcf
DETAILS
43
Stopped
OSCC5
input/OSCC5.vcf
DETAILS
1 - 10 of 27 processes
1 2 3
Process HCC2_2
Completed: Using Tomo Flow backend.
Parameters
C
Log
pVACseq
Commands
Results
HCC2_2
HCC2/HCC2_2.all_epitopes.tsv (Processed data from IEDB, but with no filtering or extra data)
X-Axis Value
Corresponding WT Score
Y-Axis Value
Best MT Score
Show 3972 results with null X or Y axis values
Binding Threshold (best): 0 .. 28650
Binding Threshold (median): 0 .. 34350
Fold Change: 0 .. 50
Tumor DNA Depth: 0 .. 305
Tumor DNA VAF: 0 .. 1.50
Download
Download Selected
D
Best MT Score vs Corresponding WT Score
Best MT Score
Corresponding WT Score
30000
25000
20000
15000
10000
5000
0
0
10000
20000
30000