pVAC-Seq Documentation

Release 4.0.9

Jasreet Hundal, Susanna Kiwala, Aaron Graubert, Jason Walker, C

Apr 24, 2017

Contents

pVAC-Seq is a cancer immunotherapy pipeline for the identification of **p**ersonalized **V**ariant **A**ntigens by **C**ancer **Seq**uencing (pVAC-Seq) that integrates tumor mutation and expression data (DNA- and RNA-Seq). It enables cancer immunotherapy research by using massively parallel sequence data to predicting tumor-specific mutant peptides (neoantigens) that can elicit anti-tumor T cell immunity. It is being used in studies of checkpoint therapy response and to identify targets for cancer vaccines and adoptive T cell therapies. For more general information, see the manuscript published in Genome Medicine.

CHAPTER 1

Features

SNV and Indel support

pVAC-Seq offers epitope binding predictions for missense, inframe indel, and frameshift mutations.

VCF support

pVAC-Seq uses a single-sample VCF file as its input. This VCF file must be annotated with VEP. See the *Prerequisites* for more information.

No local install of epitope prediction software needed

pVAC-Seq utilizes the IEDB RESTful web interface. This means that none of the underlying prediction software, like NetMHC, needs to be installed locally.

Warning: We only recommend using the RESTful API for small requests. If you use the RESTful API to process large VCFs or to make predictions for many alleles, epitope lengths, or prediction algorithms, you might overload their system. This can result in the blacklisting of your IP address by IEDB, causing 403 errors when trying to use the RESTful API. In that case please open a ticket with IEDB support to have your IP address removed from the IEDB blacklist.

Support for local installation of the IEDB Analysis Resources

pVAC-Seq provides the option of using a local installation of the IEDB MHC class I and class II binding prediction tools.

Warning: Using a local IEDB installation is strongly recommended for larger datasets or when the making predictions for many alleles, epitope lengths, or prediction algorithms. More information on how to install IEDB locally can be found on the *Installation* page.

MHC Class I and Class II predictions

Both MHC Class I and Class II predictions are supported. Simply choose the desired prediction algorithms and HLA alleles during processing and Class I and Class II prediction results will be written to their own respective subdirectories in your output directory.

By using the IEDB RESTful web interface, pVAC-Seq leverages their extensive support of different prediction algorithms.

MHC Class I Prediction Algorithm	Version
NetMHCpan	2.8
NetMHC	4.0
NetMHCcons	1.1
PickPocket	1.1
SMM	
SMMPMBEC	

MHC Class II Prediction Algorithm	Version
NetMHCIIpan	3.0
SMMalign	1.1
NNalign	2.2

Comprehensive filtering

Automatic filtering on the binding affinity ic50 value narrows down the results to only include “good” candidate peptides. The binding filter threshold can be adjusted by the user for each pVAC-Seq run, and additional filtering can be manually done by the user on the final result file to narrow down the candidate epitopes even further.

bam-readcount and cufflinks files can be provided by the user as additional input files and are used to extract coverage and expression data. When any bam-readcount or cufflinks files are provided, automatic filtering with adjustable thresholds on depth, VAF, and/or expression value will narrow down the results. The user can also manually run the coverage filter to further narrow down their results from the final output file.

The user can also specify an option to only keep the top scoring result for each allele-peptide length combination for each variant.

NetChop and NetMHCstab integration

Cleavage position predictions are added with optional processing through NetChop.

Stability predictions can be added if desired by the user. These predictions are obtained via NetMHCstab.

CHAPTER 2

Installation

pVAC-Seq is written for Linux but some users have been able to run it successfully on Mac OS X. If you are using Windows you will need to set up a Linux environment, for example by setting up a virtual machine.

pVAC-Seq requires Python 3.5. Before running any installation steps check the Python version installed on your system:

```
python -V
```

If you don't have Python 3.5 installed, we recommend using Conda to emulate a Python 3.5. environment. We've encountered problems with users that already have Python 2.x installed when they also try to install Python 3.5. The defaults will not be set correctly in that case. If you already have Python 2.x installed we **strongly** recommmend using Conda instead of installing Python 3.5 locally.

Once you have set up your Python 3.5 environment correctly you can use `pip` to install pVAC-Seq. Make sure you have `pip` installed. `pip` is generally included in python distributions, but may need to be upgraded before use. See the instructions for installing or upgrading `pip`.

After you have pip installed, type the following command on your Terminal (for Mac and Linux users) or the Command Prompt (for Windows users):

```
pip install pvacseq
```

You can check that `pvacseq` has been installed under the default environment by listing all installed packages:

```
pip list
```

`pip` will fetch and install pVAC-Seq and its dependencies for you. After installing, you can run `pvacseq` directly from the Terminal/Command Prompt.

If you have an old version of pVAC-Seq installed you might want to consider upgrading to the latest version:

```
pip install pvacseq --upgrade
```

Installing IEDB binding prediction tools (strongly recommended)

Warning: Using a local IEDB installation is strongly recommended for larger datasets or when the making predictions for many alleles, epitope lengths, or prediction algorithms.

Warning: The IEDB binding prediction tools are only compatible with Linux.

You may create a local install of the IEDB binding prediction tools by first downloading the archives for class I and class II from the IEDB website. If using both the Class I and the Class II tools, they both need to be installed into the same parent directory.

Note: IEDB requires tcsh. You can install it by running `sudo apt-get install tcsh`.

MHC Class I

Download the archives for class I and unpack them.

```
tar -zxvf IEDB_MHC_I-2.15.2.tar.gz
cd mhc_i
./configure
```

Note: Running the `configure` script requires a Python 2 environment. If you are currently emulating a Python 3 environment with Conda you will need to run `source deactivate` before executing the `configure` script.

Open `method/netmhc_4_0_executable/__init__.py` and delete/comment out the first line (`import pkg_resources`). Also delete/comment out the same line of code from `method/netmhcpan_3_0_executable/__init__.py` on line 7.

If you want to use the NetMHCcons prediction algorithm you will need to change the shebang line of certain files to explicitly use python2.7. The files in question are:

- `method/netMHCcons-1.1/bin/pseudofind`
- `method/netMHC-3.4/netMHC`

In these files change the shebang line to `#! /usr/bin/env python2.7`.

MHC Class II

```
tar -zxvf IEDB_MHC_II-2.16.tar.gz
cd mhc_ii
./configure.py
```

Note: Running the `configure` script requires a Python 2 environment. If you are currently emulating a Python 3 environment with Conda you will need to run `source deactivate` before executing the `configure` script.

Getting Started

pVAC-Seq provides a set of example data to show the expected input and output files. You can download the data set by running the `pvacseq download_example_data` *command*.

The example data output can be reproduced by running the following command:

```
pvacseq run \
<example_data_dir>/input.vcf \
Test \
HLA-G*01:09,HLA-E*01:01,H2-IAb \
NetMHC PickPocket NNalign <output_dir> \
-e 9,10 \
-i <example_data_dir>/additional_input_file_list.yaml --tdna-vaf 20 \
--net-chop-method cterm --netmhc-stab \
--top-score-metric=lowest -d full --keep-tmp-files
```

A detailed description of all command options can be found on the *Usage* page.

CHAPTER 3

Prerequisites

VEP

The input to the pVAC-Seq pipeline is a VEP annotated single-sample VCF. In addition to the standard VEP annotations, pVAC-Seq also requires the annotations provided by the Downstream and Wildtype VEP plugins.

To create a VCF for use with pVAC-Seq follow these steps:

1. Download and install the VEP command line tool following these instructions.
2. Download the VEP_plugins from their GitHub repository.
3. *Copy the Wildtype plugin* provided with the pVAC-Seq package to the folder with the other VEP_plugins:

```
pvacseq install_vep_plugin
```

4. Run VEP on the input vcf with at least the following options:

```
--format vcf
--vcf
--symbol
--plugin Downstream
--plugin Wildtype
--terms SO
```

The `--dir_plugins <VEP_plugins directory>` option may need to be set depending on where the VEP_plugins were installed to.

Additional VEP options that might be desired can be found here.

Example VEP Command

```
perl variant_effect_predictor.pl \
--input_file <input VCF> --format vcf --output_file <output VCF> \
--vcf --symbol --terms SO --plugin Downstream --plugin Wildtype \
[--dir_plugins <VEP_plugins directory>]
```

Optional Preprocessing

Coverage and expression data can be added to the pVAC-Seq processing by providing bam-readcount and/or Cufflinks output files as additional input files. These additional input files must be provided as a yaml file in the following structure:

```
gene_expn_file: <genes.fpkm_tracking file from Cufflinks>
transcript_expn_file: <isoforms.fpkm_tracking file from Cufflinks>
normal_snvs_coverage_file: <bam-readcount output file for normal BAM and snvs>
normal_indels_coverage_file: <bam-readcount output file for normal BAM and indels>
tdna_snvs_coverage_file: <bam-readcount output file for tumor DNA BAM and snvs>
tdna_indels_coverage_file: <bam-readcount output file for tumor DNA BAM and indels>
trna_snvs_coverage_file: <bam-readcount output file for tumor RNA BAM and snvs>
trna_indels_coverage_file: <bam-readcount output file for tumor RNA BAM and indels>
```

Each file in this list is optional, and its entry can be omitted. If no additional files exist then this yaml file is optional and can be omitted from the list of `pvacseq` arguments.

bam-readcount

pVAC-Seq optionally accepts bam-readcount files as inputs to add coverage information (depth and VAF) for downstream filtering. Depth and VAF are calculated from the read counts of the reference allele and alternate allele.

Follow the installation instructions on the bam-readcount GitHub page.

bam-readcount uses a bam file and regions file as input, and the bam regions may either contain snvs or indels. Indel regions must be run in a special insertion-centric mode. Any mixed input regions must be split into snvs and indels, and bam-reacount must then be run on each file individually using the same bam.

Example bam-readcount command

```
bam-readcount -f <reference fasta> -l <site list> <bam_file>
```

The `-i` option must be used when running indels bam in order to process indels in insertion-centric mode.

A minimum base quality of 20 is recommended which can be enabled by `-b 20`.

Cufflinks

pVAC-Seq optionally accepts Cufflinks files as inputs to extract gene and transcript expression data for downstream filtering.

Installation instructions for Cufflinks can be found on their GitHub page.

Example Cufflinks command

```
cufflinks <sam_file>
```

CHAPTER 4

Usage

Warning: Using a local IEDB installation is strongly recommended for larger datasets or when the making predictions for many alleles, epitope lengths, or prediction algorithms. More information on how to install IEDB locally can be found on the *Installation* page.

```
usage: pvacseq run [-h] [-e EPITOPE_LENGTH] [-l PEPTIDE_SEQUENCE_LENGTH]
                   [--iedb-install-directory IEDB_INSTALL_DIRECTORY]
                   [-i ADDITIONAL_INPUT_FILE_LIST]
                   [--net-chop-method {cterm,20s}] [--netmhc-stab] [-t]
                   [-m {lowest,median}] [-b BINDING_THRESHOLD]
                   [-c MINIMUM_FOLD_CHANGE] [--normal-cov NORMAL_COV]
                   [--tdna-cov TDNA_COV] [--trna-cov TRNA_COV]
                   [--normal-vaf NORMAL_VAF] [--tdna-vaf TDNA_VAF]
                   [--trna-vaf TRNA_VAF] [--expn-val EXPN_VAL]
                   [--net-chop-threshold NET_CHOP_THRESHOLD]
                   [-a {sample_name}] [-s FASTA_SIZE] [-r IEDB_RETRIES]
                   [-d DOWNSTREAM_SEQUENCE_LENGTH] [-k]
                   input_file sample_name allele
                   {NNalign,NetMHC,NetMHCIIpan,NetMHCcons,NetMHCpan,PickPocket,SMM,
↪SMMPMBEC,SMMalign}
                   [{NNalign,NetMHC,NetMHCIIpan,NetMHCcons,NetMHCpan,PickPocket,SMM,
↪SMMPMBEC,SMMalign} ...]
                   output_dir
```

Positional Arguments

input_file A VEP-annotated single-sample VCF containing transcript, Wildtype protein sequence, and Downstream protein sequence information

sample_name The name of the sample being processed. This will be used as a prefix for output files

allele Name of the allele to use for epitope prediction. Multiple alleles can be specified using a comma-separated list. For a list of available alleles, use: *pvacseq valid_alleles*

prediction_algorithms Possible choices: NNalign, NetMHC, NetMHCIIpan, NetMHCcons, NetMHCpan, PickPocket, SMM, SMMPMBEC, SMMalign

The epitope prediction algorithms to use. Multiple prediction algorithms can be specified, separated by spaces

output_dir The directory for writing all result files

Named Arguments

-e, --epitope-length Length of subpeptides (neoepitopes) to predict. Multiple epitope lengths can be specified using a comma-separated list. Typical epitope lengths vary between 8-11. Required for Class I prediction algorithms

-l, --peptide-sequence-length Length of the peptide sequence to use when creating the FASTA. Default: 21

Default: 21

--iedb-install-directory Directory that contains the local installation of IEDB MHC I and/or MHC II

-i, --additional-input-file-list yaml file of additional files to be used as inputs, e.g. cufflinks output files. For an example of this yaml file run *pvacseq config_files additional_input_file_list*.

--net-chop-method Possible choices: cterm, 20s

NetChop prediction method to use (“cterm” for C term 3.0, “20s” for 20S 3.0).

--netmhc-stab Run NetMHCStabPan after all filtering and add stability predictions to predicted epitopes

Default: False

-t, --top-result-per-mutation Output only the top scoring result for each allele-peptide length combination for each variant. Default: False

Default: False

-m, --top-score-metric Possible choices: lowest, median

The ic50 scoring metric to use when filtering epitopes by binding-threshold or minimum fold change. lowest: Best MT Score/Corresponding Fold Change - lowest MT ic50 binding score/corresponding fold change of all chosen prediction methods. median: Median MT Score/Median Fold Change - median MT ic50 binding score/fold change of all chosen prediction methods. Default: median

Default: “median”

-b, --binding-threshold Report only epitopes where the mutant allele has ic50 binding scores below this value. Default: 500

Default: 500

-c, --minimum-fold-change Minimum fold change between mutant binding score and wild-type score. The default is 0, which filters no results, but 1 is often a sensible choice (requiring that binding is better to the MT than WT). Default: 0

Default: 0

--normal-cov Normal Coverage Cutoff. Sites above this cutoff will be considered. Default: 5

Default: 5

--tdna-cov Tumor DNA Coverage Cutoff. Sites above this cutoff will be considered. Default: 10

Default: 10

--trna-cov Tumor RNA Coverage Cutoff. Sites above this cutoff will be considered. Default: 10

Default: 10

--normal-vaf Normal VAF Cutoff. Sites BELOW this cutoff in normal will be considered. Default: 2

Default: 2

--tdna-vaf Tumor DNA VAF Cutoff. Sites above this cutoff will be considered. Default: 40

Default: 40

--trna-vaf Tumor RNA VAF Cutoff. Sites above this cutoff will be considered. Default: 40

Default: 40

--expn-val Gene and Transcript Expression cutoff. Sites above this cutoff will be considered. Default: 1

Default: 1

--net-chop-threshold NetChop prediction threshold. Default: 0.5

Default: 0.5

-a, --additional-report-columns Possible choices: sample_name

Additional columns to output in the final report

-s, --fasta-size Number of fasta entries per IEDB request. For some resource-intensive prediction algorithms like Pickpocket and NetMHCpan it might be helpful to reduce this number. Needs to be an even number.

Default: 200

-r, --iedb-retries Number of retries when making requests to the IEDB RESTful web interface. Must be less than or equal to 100.Default: 5

Default: 5

-d, --downstream-sequence-length Cap to limit the downstream sequence length for frameshifts when creating the fasta file. Use ‘full’ to include the full downstream sequence. Default: 1000

Default: “1000”

-k, --keep-tmp-files Keep intermediate output files. This migt be useful for debugging purposes.

Default: False

CHAPTER 5

Filtering Commands

pVAC-Seq currently offers two filters: a binding filter and a coverage filter.

The binding filter is always run automatically as part of the pVAC-Seq pipeline. The coverage filter is run automatically if bam-readcount or cufflinks file are provided as additional input files to a pVAC-Seq run.

Both filters can also be run manually to narrow the final results down further.

Binding Filter

The binding filter filters out variants that don't pass the chosen binding threshold. The user can chose whether to apply this filter to the "lowest" or the "median" binding affinity score. The "lowest" binding affinity score is recorded in the "Best MT Score" column and represents the lowest ic50 score of all prediction algorithms that were picked during the previous pVAC-Seq run. The "median" binding affinity score is recorded in the "Median MT Score" column and corresponds to the median ic50 score of all prediction algorithms used to create the report.

The binding filter also offers the option to filter on Fold Change columns, which contain the ratio of the MT score to the WT Score. If the binding filter is set to "best", the "Corresponding Fold Change" column will be used. ("Corresponding WT Score"/"Best MT Score"). If the binding filter is set to "median", the "Median Fold Change" column will be used ("Median WT Score"/"Median MT Score").

```
usage: pvacseq binding_filter [-h] [-b BINDING_THRESHOLD]
                              [-c MINIMUM_FOLD_CHANGE] [-m {lowest,median}]
                              input_file output_file
```

Positional Arguments

input_file The final report .tsv file to filter

output_file Output .tsv file containing list of filtered epitopes based on binding affinity

Named Arguments

-b, --binding-threshold Report only epitopes where the mutant allele has ic50 binding scores below this value. Default: 500

Default: 500

-c, --minimum-fold-change Minimum fold change between mutant binding score and wild-type score. The default is 0, which filters no results, but 1 is often a sensible option (requiring that binding is better to the MT than WT). Default: 0

Default: 0

-m, --top-score-metric Possible choices: lowest, median

The ic50 scoring metric to use when filtering epitopes by binding-threshold or minimum fold change. lowest: Best MT Score/Corresponding Fold Change - lowest MT ic50 binding score/corresponding fold change of all chosen prediction methods. median: Median MT Score/Median Fold Change - median MT ic50 binding score/fold change of all chosen prediction methods. Default: median

Default: "median"

Coverage Filter

If a pVAC-Seq process has been run with bam-readcount or Cufflinks input files then the coverage_filter can be run again on the final report file to narrow down the results even further.

If no additional coverage input files have been provided to the main pVAC-Seq run then this information would need to be manually added to the report in order to run this filter.

```
usage: pvacseq coverage_filter [-h] [--normal-cov NORMAL_COV]
                               [--tdna-cov TDNA_COV] [--trna-cov TRNA_COV]
                               [--normal-vaf NORMAL_VAF] [--tdna-vaf TDNA_VAF]
                               [--trna-vaf TRNA_VAF] [--expn-val EXPN_VAL]
                               input_file output_file
```

Positional Arguments

input_file The final report .tsv file to filter

output_file Output .tsv file containing list of filtered epitopes based on coverage and expression values

Named Arguments

--normal-cov Normal Coverage Cutoff. Sites above this cutoff will be considered. Default: 5

Default: 5

--tdna-cov Tumor DNA Coverage Cutoff. Sites above this cutoff will be considered. Default: 10

Default: 10

--trna-cov Tumor RNA Coverage Cutoff. Sites above this cutoff will be considered. Default: 10

Default: 10

--normal-vaf Normal VAF Cutoff. Sites BELOW this cutoff in normal will be considered. Default: 2

Default: 2

--tdna-vaf Tumor DNA VAF Cutoff. Sites above this cutoff will be considered. Default: 40

Default: 40

--trna-vaf Tumor RNA VAF Cutoff. Sites above this cutoff will be considered. Default: 40

Default: 40

--expn-val Gene and Transcript Expression cutoff. Sites above this cutoff will be consideredDefault: 1

Default: 1

CHAPTER 6

Additional Commands

To make using pVAC-Seq easier several convenience methods are included in the package.

Download Example Data

```
usage: pvacseq download_example_data [-h] destination_directory
```

Positional Arguments

destination_directory Directory for downloading example data

Install VEP Plugin

```
usage: pvacseq install_vep_plugin [-h] vep_plugins_path
```

Positional Arguments

vep_plugins_path Path to your VEP_plugins directory

List Valid Alleles

```
usage: pvacseq valid_alleles [-h]
                             [-p {NNalign,NetMHC,NetMHCIIpan,NetMHCcons,NetMHCpan,
↪PickPocket,SMM,SMMPMBEC,SMMalign}]
```

Named Arguments

-p, --prediction-algorithm Possible choices: NNalign, NetMHC, NetMHCIIpan, NetMHCcons, NetMHCpan, PickPocket, SMM, SMMPMBEC, SMMalign

The epitope prediction algorithms to use

Documentation For Configuration Files

```
usage: pvacseq config_files [-h] {additional_input_file_list}
```

Positional Arguments

config_file_type Possible choices: additional_input_file_list

The config file type to retrieve more information for

CHAPTER 7

Optional Downstream Analysis Tools

Generate Protein Fasta

```
usage: pvacseq generate_protein_fasta [-h] [-d DOWNSTREAM_SEQUENCE_LENGTH]
                                      input_file peptide_sequence_length
                                      output_file
```

Positional Arguments

input_file A VEP-annotated single-sample VCF containing transcript, Wildtype protein sequence, and Downstream protein sequence information

peptide_sequence_length Length of the peptide sequence to use when creating the FASTA.

output_file The output fasta file

Named Arguments

-d, --downstream-sequence-length Cap to limit the downstream sequence length for frameshifts when creating the fasta file. Use ‘full’ to include the full downstream sequence. Default: 1000

Default: “1000”

CHAPTER 8

Contact

Bug reports or feature requests can be submitted on the pVAC-Seq Github page. You may also contact us by email at pvacseq-support@gowustl.onmicrosoft.com.

CHAPTER 9

New in version 4.0.9

This release adds handling for DNPs and MNPs missense mutations.

This version adds a new option `--additonal-report-columns` to the `pvacseq run` command which can be use to append additional columns of data to the report. Right now the only value supported for this option is `sample_name` which appends a column with the sample name to the final report.

We updated the logic that determines whether or not a corresponding wildtype epitope for a mutant epitope is included in the report. Previously, we would only include the corresponding wildtype epitope if the number of **consecutive** matching amino acids between mutant and wildtype epitope was larger then half of the total number of amino acids in the epitope. We now use the **total** number of matching amino acids between the mutant epitope and the corespondig wildtype epitope across the whole length of the epitope to make that determination. The total number of matching amino acids needs to be larger than half of the length of the epitope. Otherwise the corresponding wildtype epitope is reported as “NA”.

With this release any execution of `pvacseq run` will create a log file of the inputs used. This log file is then used when executing another run with the same output directory. This ensures that you can only write to the same output directory if identical parameters are used.

CHAPTER 10

Citation

Jasreet Hundal, Beatriz M. Carreno, Allegra A. Petti, Gerald P. Linette, Obi L. Griffith, Elaine R. Mardis, and Malachi Griffith. pVAC-Seq: A genome-guided in silico approach to identifying tumor neoantigens. Genome Medicine. 2016, 8:11, DOI: 10.1186/s13073-016-0264-5. PMID: 26825632.

CHAPTER 11

License

This project is licensed under NPOSL-3.0.