pVAC-Seq Documentation

Release 4.0.10

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

Sep 05, 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
```

2.1 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`.

2.1.1 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`.

2.1.2 MHC Class II

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

Open the `configure.py` file and update the lines that set the `smm` and `nn` variables to use relative paths like so:

```
smm = re.compile(curDir + "/netMHCII-1.1")
nn = re.compile(curDir + "/netMHCII-2.2")
```

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.

2.2 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

3.1 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.

The `--pick` option might be useful to limit the annotation to the top transcripts. Otherwise, VEP will annotate each variant with all possible transcripts. pVAC-Seq will provide predictions for all transcripts in the VEP CSQ field. Running VEP without the `--pick` option can therefor drasticly increase the runtime of pVAC-Seq.

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>]
```

3.2 Optional Preprocessing

3.2.1 Coverage and Expression Data

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>
```

You may also provide FPKM values from other sources by creating cufflinks-formatted input files.

For transcript FPKM: a tab-separated file with a `tracking_id` column containing Ensembl transcript IDs and a `FPKM` column containing FPKM values.

For gene FPKM: a tab-separated file with a `tracking_id` column containing Ensembl gene IDs, a `locus` column describing the region within the gene, and a `FPKM` column containing FPKM values. In the pVAC-Seq pipeline the FPKM values will be summed for all loci of a gene. You may also provide already summed FPKM values. In that case you will still need to provide a `locus` column but the values in that column can be empty.

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.

For usage instructions run

```
pvacseq run --help
```

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.

5.1 Binding Filter

For usage instructions run

```
pvacseq binding_filter --help
```

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").

5.2 Coverage Filter

For usage instructions run

```
pvacseq coverage_filter --help
```

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.

CHAPTER 6

Additional Commands

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

6.1 Download Example Data

For usage instructions run

```
pvacseq download_example_data --help
```

6.2 Install VEP Plugin

For usage instructions run

```
pvacseq install_vep_plugin --help
```

6.3 List Valid Alleles

For usage instructions run

```
pvacseq valid_alleles --help
```

6.4 Documentation For Configuration Files

For usage instructions run

```
pvacseq config_files --help
```

CHAPTER 7

Optional Downstream Analysis Tools

7.1 Generate Protein Fasta

For usage instructions run

```
pvacseq generate_protein_fasta --help
```

CHAPTER 8

Frequently Asked Questions

What type of variants does pVAC-Seq support?

pVAC-Seq makes predictions for all transcripts of a variant that were annotated as `missense_variant`, `inframe_insertions`, `inframe_deletion`, or `frameshift_variant` by VEP as long as the transcript was not also annotated as `start_lost`. In addition, pVAC-Seq only includes variants that were called as homozygous or heterozygous variant. Variants that were not called are skipped.

My pVAC-Seq command has been running for a long time. Why is that?

The rate-limiting factor in running pVAC-Seq is the number of calls that are made to the IEDB software for binding score predictions.

Note: It is generally faster to make IEDB calls using a local install of IEDB than using the IEDB web API. It is, therefore, recommended to use a local IEDB install for any in-depth analysis.

There are a number of factors that determine the number of IEDB calls to be made:

- Number of variants in your VCF

  pVAC-Seq will make predictions for each missense, inframe indel, and frameshift variant in your VCF.

  Speedup suggestion: Split the VCF into smaller subsets and process each one individually, in parallel.

- Number of transcripts for each variant

  pVAC-Seq will make predictions for each transcript of a supported variant individually. The number of transcripts for each variant depends on how VEP was run when the VCF was annotated.

  Speedup suggestion: Use the `--pick` option when running VEP to annotate each variant with the top transcript only.

- The `--fasta-size` parameter value

  pVAC-Seq takes an input VCF and creates a wildtype and a mutant fasta for each transcript. The number of fasta entries that get submitted to IEDB at a time is limited by the `--fasta-size` parameter in order to reduce the load on the IEDB servers. The smaller the fasta-size, the more calls have to be made to IEDB.

Speedup suggestion: When using a local IEDB install, increase the size of this parameter.

- Number of prediction algorithms, epitope lengths, and HLA-alleles

  One call to IEDB is made for each combination of these parameters for each chunk of fasta sequences. That means, for example, when 7 prediction algorithms, 4 epitope lengths, and 6 HLA-alleles are chosen, 7*4*6=168 calls to IEDB have to be made for each chunk of fastas.

  Speedup suggestion: Reduce the number of prediction algorithms, epitope lengths, and/or HLA-alleles to the ones that will be the most meaningful for your analysis. For example, the NetMHCcons method is already a consensus method between NetMHC, NetMHCpan, and PickPocket. If NetMHCcons is chosen, you may want to omit the underlying prediction methods. Likewise, if you want to run NetMHC, NetMHCpan, and PickPocket individually, you may want to skip NetMHCcons.

- `--downstream-sequence-length` parameter value

  This parameter determines how many amino acids of the downstream sequence after a frameshift mutation will be included in the wildtype fasta sequence. The shorter the downstream sequence length, the lower the number of epitopes that IEDB needs to make binding predictions for.

  Speedup suggestion: Reduce the value of this parameter.

My pVAC-Seq output file does not contain entries for all of the alleles I chose. Why is that?

There could be a few reasons why the pVAC-Seq output does not contain predictions for alleles:

- The alleles you picked might've not been compatible with the prediction algorithm and/or epitope lengths chosen. In that case no calls for that allele would've been made and a status message would've printed to the screen.
- It could be that all epitope predictions for some alleles got filtered out. You can check the `<sample_name>.combined.parsed.tsv` file to see all called epitopes before filtering.

Why are some values in the WT Epitope Seq column NA ?

Not all mutant epitope sequences will have a corresponding wildtype epitope sequence. This occurs when the mutant epitope sequence is novel and a comparison is therefore not meaningful:

- An epitope in the downstream portion of a frameshift might not have a corresponding wildtype epitope at the same position at all. The epitope is completely novel.
- An epitope that overlaps an inframe indel or multinucleotide polymorphism (MNP) might have a large number of amino acids that are different from the wildtype epitope at the corresponding position. If less than half of the amino acids between the mutant epitope sequence and the corresponding wildtype sequence match, the corresponding wildtype sequence in the report is set to `NA`.

What filters are applied during a pVAC-Seq run?

By default we filter the neoepitopes on their binding score. If bam-readcount files and/or cufflinks files are provided we also filter on the depth, VAF, and FPKM. In addition, candidates where the mutant epitope sequence is the same as the wildtype epitope sequence will also be filtered out.

How can I see all of the candidate epitopes without any filters applied?

The `<sample_name>.combined.parsed.tsv` will contain all of the epitopes predicted before filters are applied.

Why have some of my epitopes been filtered out even though the Best MT Score is below 500?

By default, the binding filter will be applied to the `Median MT Score` column. This is the median score value among all chosen prediction algorithms. The `Best MT Score` column shows the lowest score among all chosen prediction algorithms. To change this behavior and apply the binding filter to the `Best MT Score` column you may set the `--top-score-metric` parameter to `lowest`.

Why are entries with NA in the VAF and depth columns not filtered?

We do not filter out `NA` entries for depth and VAF since there is not enough information to determine whether the cutoff has been met one way or another.

Why don't some of my epitopes have score predictions for certain prediction methods?

Not all prediction methods support all epitope lengths or all alleles. To see a list of supported alleles for a prediction method you may use the `pvacseq valid_alleles` *command*. For more details on each algorithm refer to the IEDB MHC Class I and Class II documentation.

How do I use StringTie instead of Cufflinks for transcript/gene abundance estimates?

You may also provide FPKM values from other sources, including StringTie, by creating cufflinks-formatted input files.

For transcript FPKM: a tab-separated file with a `tracking_id` column containing Ensembl transcript IDs and a `FPKM` column containing FPKM values.

For gene FPKM: a tab-separated file with a `tracking_id` column containing Ensembl gene IDs, a `locus` column describing the region within the gene, and a `FPKM` column containing FPKM values. In the pVAC-Seq pipeline the FPKM values will be summed for all loci of a gene. You may also provide already summed FPKM values. In that case you will still need to provide a `locus` column but the values in that column can be empty.

How is pVAC-Seq licensed?

pVAC-Seq is licensed under NPOSL-3.0.

How do I cite pVAC-Seq?

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 9

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 10

New in version 4.0.10

This is a hotfix release to fix a bug with how certain types of frameshift mutations were handled. Previously, we were not creating the correct mutant peptide sequence for these variants. See this GitHub issue for more information.

This version also includes a sanity check to error out if the wildtype amino acid in the wildtype protein sequence differs from the expected wildtype amino acid as listed in the protein change. This situation might occur if the VCF was annotated with a different reference build than the one used for alignment and variant calling.

CHAPTER 11

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 12

License

This project is licensed under NPOSL-3.0.