Vaxrank: A computational tool for designing personalized cancer vaccines

Alexander Rubinsteyn[†,1], Isaac Hodes[1], Julia Kodysh[1], Jeffrey Hammerbacher[1,2]

[1] Department of Genetics and Genomic Sciences at Icahn School of Medicine at Mount Sinai
[2] Department of Microbiology and Immunology, Medical University of South Carolina
[†] Contact: alex@hammerlab.org

Abstract

Therapeutic vaccines targeting mutant tumor antigens ("neoantigens") are an increasingly popular form of personalized cancer immunotherapy. Vaxrank is a computational tool for selecting neoantigen vaccine peptides from tumor mutations, tumor RNA data, and patient HLA type. Vaxrank is freely available at www.github.com/hammerlab/vaxrank under the Apache 2.0 open source license and can also be installed from the Python Package Index.

1 Introduction

Mutated cancer proteins recognized by T-cells have become known as "neoantigens" and are considered an essential component of a tumor-specific immune response (Finnigan *et al.*, 2015; Gubin *et al.*, 2015; Schumacher and Schreiber, 2015). Therapeutic vaccination against neoantigens is an emerging experimental cancer therapy that attempts to mobilize an antigen-specific immune response against mutated tumor proteins (Türeci *et al.*, 2016; Zhang *et al.*, 2017). Since few tumor mutations are shared between patients, neoantigen vaccines must be personalized therapies. A common approach for achieving personalization is high-throughput sequencing of tumor and normal patient samples followed by in-silico prioritization of mutated peptides that are likely to be presented on the surface of tumor cells by MHC (major histocompatibility complex) molecules.

Vaxrank is a tool for selecting mutated peptides for personalized therapeutic cancer vaccination. Vaxrank determines which peptides should be used in a vaccine from tumor-specific somatic mutations, tumor RNA sequencing data, and a patient's HLA type. These peptides can then be synthesized and combined with an adjuvant to attempt to elicit an anti-tumor T-cell response in a patient.

The sequence of each mutated protein is determined by assembling variant RNA reads. Mutant protein sequences are ranked using a scoring system which seeks to satisfy two objectives: choosing mutations that are abundant in the tumor and choosing those whose translated amino acid sequences contain likely MHC ligands. Additionally, Vaxrank considers surrounding non-mutated residues in a peptide to prioritize vaccine peptide candidates and to improve the odds of successful synthesis.

Vaxrank was designed for and is currently being used in the Personalized Genomic Vaccine Phase I clinical trial at the Icahn School of Medicine at Mount Sinai (NCT02721043) (Rubinsteyn *et al.*, 2016a).
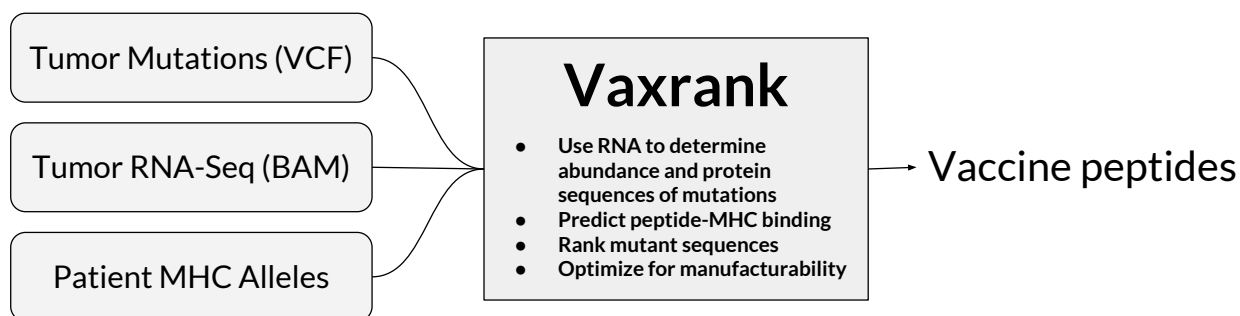
Figure 1: Users provide tumor mutations, tumor RNA sequence data, and patient HLA type. These are used to determine mutant protein sequences and rank them according to expression and predicted MHC affinity.

2   Running Vaxrank

To generate a Vaxrank vaccine report, the user must provide one or more files containing somatic variants (in VCF, MAF, or JSON format), aligned tumor RNA-seq reads (as an indexed BAM), and the HLA alleles to be used for MHC binding prediction:

```
vaxrank
   --vcf somatic-variants.vcf
   --bam tumor-rna.bam
   --mhc-predictor netmhc
   --mhc-alleles H2-Kb,H2-Db
   --mhc-peptide-lengths 8-10
   --vaccine-peptide-length 21
   --min-alt-rna-reads 3
   --output-pdf-report vaccine-peptides.pdf
```

The `--mhc-predictor` argument controls which program is used to predict the affinity between a peptide-MHC pair. Vaxrank supports the use of locally installed instances of NetMHC (Andreatta and Nielsen, 2016), NetMHCpan (Nielsen *et al.*, 2007), NetMHCcons (Karosiene *et al.*, 2012), MHCflurry (Rubinsteyn *et al.*, 2016b), or a variety of web-based predictors through IEDB (Vita *et al.*, 2015). The `--min-alt-rna-reads` argument controls the minimum number of RNA reads supporting a variant required to include that variant in the output report. In addition to quantifying tumor expression of a mutations, the RNA reads are used to phase adjacent variants when reconstructing the mutated coding sequence. A more complete list of options for input data, filtering, and output formats can be seen by running `vaxrank --help`. Vaxrank's output can be formatted as PDF, plain-text, HTML, or an Excel spreadsheet. The output lists variants in ranked order along with vaccine peptide(s) containing that variant, predicted MHC ligands, number of supporting RNA reads, and sequence properties that affect manufacturability.

3   Ranking Mutations

A patient's coding mutations are ranked according to a score that combines each mutation's degree of expression and aggregate affinity of overlapping mutant peptides for that patient's MHC alleles.

$$RankingScore = ExpressionScore \cdot TotalBindingScore$$
$$ExpressionScore = \sqrt{\# \ RNA \ reads \ supporting \ variant \ allele}$$
$$TotalBindingScore = \sum_{s \in subsequences} \sum_{a \in alleles} BindingScore(s, a)$$

The *BindingScore* function is, by default, a logistic transformation of the peptide-MHC binding affinity that loosely approximates the probability of T-cell response (Sette *et al.*, 1994). Alternatively, binding predictions can be scored using an affinity threshold (commonly ≤ 500nM) or a threshold on the percentile rank of the affinity. Only subsequences which overlap mutant residues and do not occur in the reference proteome are considered as part of the *TotalBindingScore*.

4   Manufacturability

Vaxrank was designed under the assumption that its output will be used to make long peptides, due to their favorable immunological properties (Rosalia *et al.*, 2013). Unfortunately, long peptides are also more difficult to synthesize using traditional solid phase chemistry (Bodanszky, 1988). To avoid known difficulties in synthesis, Vaxrank selects a window of amino acids around each mutation that minimizes the following undesirable properties:

1. total number of cysteine residues
2. $max(0,$ mean hydrophobicity of 7 residues at C-terminus)
3. $max(0,$ mean hydrophobicity of any 7 amino acid window)
4. glutamine, glutamic acid, or cysteine at N-terminus
5. cysteine at C-terminus
6. proline at C-terminus
7. asparagine at N-terminus
8. total number of asparagine-proline bonds

Manufacturability optimization does not affect the ranking of mutations but is only used for selecting which surrounding residues should be included. In cases where a mutation spans a "difficult" sequence (e.g. long hydrophobic stretch), minimizing these criteria may fail to salvage manufacturability.

References

Andreatta, M. and Nielsen, M. (2016). Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics*, **32**(4), 511–517.

Bodanszky, P. D. M. (1988). *Peptide Chemistry: A Practical Textbook*. Springer Berlin Heidelberg.

Finnigan, Jr, J. P. *et al.* (2015). Mutation-Derived tumor antigens: Novel targets in cancer immunotherapy. *Oncology*, **29**(12).

Gubin, M. M. *et al.* (2015). Tumor neoantigens: building a framework for personalized cancer immunotherapy. *J. Clin. Invest.*, **125**(9), 3413–3421.

Karosiene, E. *et al.* (2012). NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. *Immunogenetics*, **64**(3), 177–186.

Nielsen, M. *et al.* (2007). NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. *PLoS One*, **2**(8), e796.

Rosalia, R. A. *et al.* (2013). Dendritic cells process synthetic long peptides better than whole protein, improving antigen presentation and t-cell activation. *Eur. J. Immunol.*, **43**(10), 2554–2565.

Rubinsteyn, A. *et al.* (2016a). Abstract a022: Computational pipeline for a personalized genomic vaccine trial. *Cancer Immunol Res*, **4**(11 Supplement), A022–A022.

Rubinsteyn, A. *et al.* (2016b). Predicting peptide-mhc binding affinities with imputed training data. *bioRxiv*, page 054775.

Schumacher, T. N. and Schreiber, R. D. (2015). Neoantigens in cancer immunotherapy. *Science*, **348**(6230), 69–74.

Sette, A. *et al.* (1994). The relationship between class I binding affinity and immunogenicity of potential cytotoxic T cell epitopes. *J. Immunol.*, **153**(12), 5586–5592.

Türeci, Ö. *et al.* (2016). Targeting the heterogeneity of cancer with individualized neoepitope vaccines. *Clin. Cancer Res.*, **22**(8), 1885–1896.

Vita, R. *et al.* (2015). The immune epitope database (IEDB) 3.0. *Nucleic Acids Res.*, **43**(Database issue), D405–12.

Zhang, X. *et al.* (2017). Personalized cancer vaccines: Targeting the cancer mutanome. *Vaccine*, **35**(7), 1094–1100.

<div align="center">

Supplemental Materials

</div>

Installing Vaxrank

Vaxrank can be installed using pip:

```
pip install vaxrank
```

This will install the Vaxrank library, along with all of its dependencies, including PyEnsembl, Varcode, Topiary, and MHCtools.

To generate PDF reports, you first need to install wkhtmltopdf. On Mac OS X this can be done by running:

```
brew install Caskroom/cask/wkhtmltopdf
```

Vaxrank uses PyEnsembl for accessing information about the reference genome. You must install an Ensembl release corresponding to the reference genome associated with the mutations provided to Vaxrank.

The latest supported release for GRCh38 is Ensembl 87:

```
pyensembl install --release 87 --species human
```

The last release for GRCh37 was Ensembl 75:

```
pyensembl install --release 75 --species human
```

If using Vaxrank for a large number of mutations, it is recommended to locally install an MHC binding predictor such as NetMHCpan or MHCflurry. NetMHCpan can be downloaded from www.cbs.dtu.dk/services/NetMHCpan/, whereas MHCflurry can be installed by running the following commands:

```
pip install mhcflurry
mhcflurry-downloads fetch
```

Basic Usage

```
vaxrank \
--vcf somatic-variants.vcf \
--bam tumor-rna.bam \
--mhc-predictor netmhc \
--mhc-alleles A*02:01,A*02:03 \
--mhc-epitope-lengths 8 \
--padding-around-mutation 5 \
--vaccine-peptide-length 25 \
--output-ascii-report vaccine-peptides-report.txt
```

This tells Vaxrank to:

- load mutations from the input VCF file `somatic-variants.vcf`

- look for evidence of expression for mutations in the RNA eBAM file `tumor-rna.bam`

- predict MHC binding of each possible 8mer peptide overlapping expressed mutations using the NetMHC prediction algorithm with the A*02:01 and A*02:03 MHC alleles

- choose protein vaccine candidates composed of 25 amino acids

- write top ranked variants with their associated vaccine proteins to `vaccine-peptides-report.txt`

You can read the complete Vaxrank documentation at vaxrank.readthedocs.io.