Methods in
Molecular Biology 2959

Springer Protocols

Sweta Rani
Lukasz Skalniak *Editors*

IMMUNO-model in Cancer

Methods and Protocols

cost
EUROPEAN COOPERATION
IN SCIENCE & TECHNOLOGY

Funded by
the European Union

OPEN ACCESS

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, UK

For further volumes:
http://www.springer.com/series/7651

For over 35 years, biological scientists have come to rely on the research protocols and methodologies in the critically acclaimed *Methods in Molecular Biology* series. The series was the first to introduce the step-by-step protocols approach that has become the standard in all biomedical protocol publishing. Each protocol is provided in readily-reproducible step-by-step fashion, opening with an introductory overview, a list of the materials and reagents needed to complete the experiment, and followed by a detailed procedure that is supported with a helpful notes section offering tips and tricks of the trade as well as troubleshooting advice. These hallmark features were introduced by series editor Dr. John Walker and constitute the key ingredient in each and every volume of the *Methods in Molecular Biology* series. Tested and trusted, comprehensive and reliable, all protocols from the series are indexed in PubMed.

IMMUNO-model in Cancer

Methods and Protocols

Edited by

Sweta Rani

Department of Science, South East Technological University, Waterford, Cork, Ireland

Lukasz Skalniak

Department of Organic Chemistry, Jagiellonian University, Kraków, Poland

Humana Press

Editors
Sweta Rani
Department of Science
South East Technological University
Waterford, Cork, Ireland

Lukasz Skalniak
Department of Organic Chemistry
Jagiellonian University
Kraków, Poland

European Cooperation in Science and Technology

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-0716-4733-2 ISBN 978-1-0716-4734-9 (eBook)
https://doi.org/10.1007/978-1-0716-4734-9

This Humana imprint is published by the registered company Springer Science+Business Media, LLC, part of Springer Nature.
The registered company address is: 1 New York Plaza, New York, NY 10004, U.S.A.

Preface

Immuno-model is a model invented to study the immune system. This book describes the computational and experimental models that help researchers understand the responses of the immune system in cancer and test experimental immuno-oncology approaches.

Macrophages can adapt to different phenotypes in response to signals from the microenvironment. This book on immuno-model describes methods to profile polarization in macrophages using ELISA. ELISA is widely used in immunology to detect proteins, antibodies, antigens, or hormones in a sample, including immune checkpoint inhibitors. Immune checkpoints are regulatory molecules that control the activation and intensity of immune responses. Glycosylation is the addition of carbohydrate groups to the proteins. Glycosylation of the immune checkpoint proteins not only promotes immune evasion in tumor cells but also holds therapeutic implications. New immune checkpoint inhibitors are warranted for better cancer treatment. One of the chapters evaluates immune checkpoint inhibitors. One of the most studied immune checkpoints is PD-1/PD-L1. There is one chapter detailing the interaction of PD-1/PD-L1 and T cells. Immune checkpoint inhibitors have significantly improved survival rates in cancers but there are patients who do not respond to these treatments. Induction of immunogenic cell death is another therapeutic option for cancer patients.

A lot of research is underway to study tumor immune microenvironment. There are several well-established in vitro models to study interaction between immune cells and cancer cells and these in vitro models are still evolving. Cells can be co-cultured using cell culture inserts or can be grown as 3D spheroids. 3D co-culture model can be used to study the interaction of immune cells and cancer cells to mimic the in vitro microenvironment. Cells can be grown in 3D using different techniques, and one of the techniques is using scaffolds derived from cancer patients. One of the chapters explores immunocompetent preclinical mouse models to study primary and metastatic brain cancer. CAR T-cell therapy is still evolving, and one of the chapters describes the method to generate CAR T-cells.

Bioinformatics has vast applications and plays a central role in immunology by enabling the analysis of large-scale datasets. Deconvolution analysis can be used to study the epigenetic dysregulation in human tumors and the tumor ecosystem. Computational methods can also be used to study the mutated peptides called neoantigens. Bioinformatics allows us to identify therapeutic targets and develop precision immunotherapies.

Waterford, Cork, Ireland — *Sweta Rani*
Kraków, Poland — *Lukasz Skalniak*

Chapter 17

Computational Methods for Cancer Neoantigen Prediction

Andrea Moreno-Manuel, Sotiris Ouzounis, Marius Eidsaa, Roberto Fornelino-González, Pilar Ballesteros-Cuartero, Daniel Gómez-Garrido, Esteban Veiga-Chacón, Theodora Katsila, Maurizio Callari, Arrate Muñoz-Barrutia, and Rebeca Sanz-Pamplona

Abstract

Neoantigens are mutated peptides arising from tumor genomic alterations, which can be recognized and attacked by the immune system, leading to antitumor immune responses. In the last decades, many immunotherapeutic strategies have been developed, which has increased the interest in neoantigens. This led to the development of computational tools that facilitate neoantigen identification and prioritization, prior to their validation using experimental approaches. This chapter aims at explaining the key steps that need to be conducted to identify potential neoantigens in silico, including an example of the most frequently used tools. This is followed by a description and comparison of the cutting-edge tools and pipelines for neoantigen prediction both for human and mouse. The last aim of this chapter is to depict the technical challenges that limit neoantigen prediction using bioinformatics, as well as the expected improvements, given the current revolution of artificial intelligence, which is implemented in most of the neoantigen-related tools. As exposed in this book chapter, we believe that advances in immunomics and computational biology will be key to implement personalized cancer immunotherapy in the clinical practice, to improve outcomes of cancer patients.

Key words Neoantigen prediction, Bioinformatics, HLA-binding affinity, MHC, Mice, Immunomics, Immune microenvironment, Cancer

1 Introduction

Over the last decades, the emergence of immunotherapy has revolutionized cancer treatment and has offered new opportunities for precise and personalized interventions. Among others, one immunotherapy strategy is the identification and targeting of tumor-specific antigens (TSAs) including neoantigens, which are peptides resulting from genetic alterations. Aberrant proteins in tumors are

Andrea Moreno-Manuel, Sotiris Ouzounis and Marius Eidsaa have contributed equally to this work.

Maurizio Callari, Arrate Muñoz-Barrutia and Rebeca Sanz-Pamplona are senior co-authors of this work.

Sweta Rani and Lukasz Skalniak (eds.), *IMMUNO-model in Cancer: Methods and Protocols*, Methods in Molecular Biology, vol. 2959, https://doi.org/10.1007/978-1-0716-4734-9_17,

degraded by the proteasome and resulting peptides are transported to the endoplasmic reticulum (ER), where they are subsequently loaded onto major histocompatibility complex (MHC) molecules, known as human leukocyte antigens (HLAs) in humans [1]. There are a variety of sources of neoantigens. Although somatic mutations, especially missense (which change the amino acid codon), are the most studied; increasing evidence supports that neoantigens can also be derived from other events such as insertion/deletions of nucleotides (INDELs), frameshift mutations (insertion or deletion of a number of nucleotides not multiple of three, thus disrupting the reading frame), gene fusions (caused by joining parts of two different genes, leading to a new protein), endogenous retroviruses (ERVs) (ERV transcripts can be a source of tumor-specific neoantigens), RNA splicing anomalies (alternative splicing consists of different exon combinations, leading to proteins with different structure and function), post-transcriptional frameshift (e.g., ribosomal slippage) or post-translational frameshift (e.g., protein splicing) [2]. In fact, the more different the neoantigen versus the wild type, the more immunogenic [3].

Neoantigens are expressed in tumors but not in healthy tissues, thus they induce stronger effector responses than tumor-associated antigens (TAAs), which are overexpressed in tumor cells but also present at a lesser extent in nonmalignant cells [4]. To trigger antitumor immune responses, neoantigens need to be presented by MHC molecules. MHC Class I molecules primarily exhibits small protein fragments derived from degraded intracellular proteins, and its role in cancer neoantigen presentation is well established. On the contrary, MHC Class II molecules exhibit extracellular antigens typically captured by antigen-presenting cells (APCs). However, MHC class II pathway is also essential for effective immune responses against neoantigens since APCs can uptake neoantigens from dying cancer cells [5, 6]. Hence, the resulting neoantigen-MHC complexes are formed and transported to the surface of cancer cells to be recognized as nonself by T-cell receptor (TCR), leading to antitumor immune responses. This specific recognition allows the elimination of malignant cells without affecting healthy tissue [7] (Fig. 1a).

Traditionally, the discovery of neoantigens has relied on experimental approaches, which make the process tedious, thus offering limited results. However, the advances in computational biology and bioinformatics, such as the use of artificial intelligence and deep learning algorithms on next-generation sequencing (NGS) data, enable a possible strategy to predict potential neoantigens faster with high accuracy [4] (Fig. 1b). Optimal pipelines discussed below not only take into account the binding capacity of peptides to MHC but also the expression levels of the antigen of interest by tumor cells [8, 9].

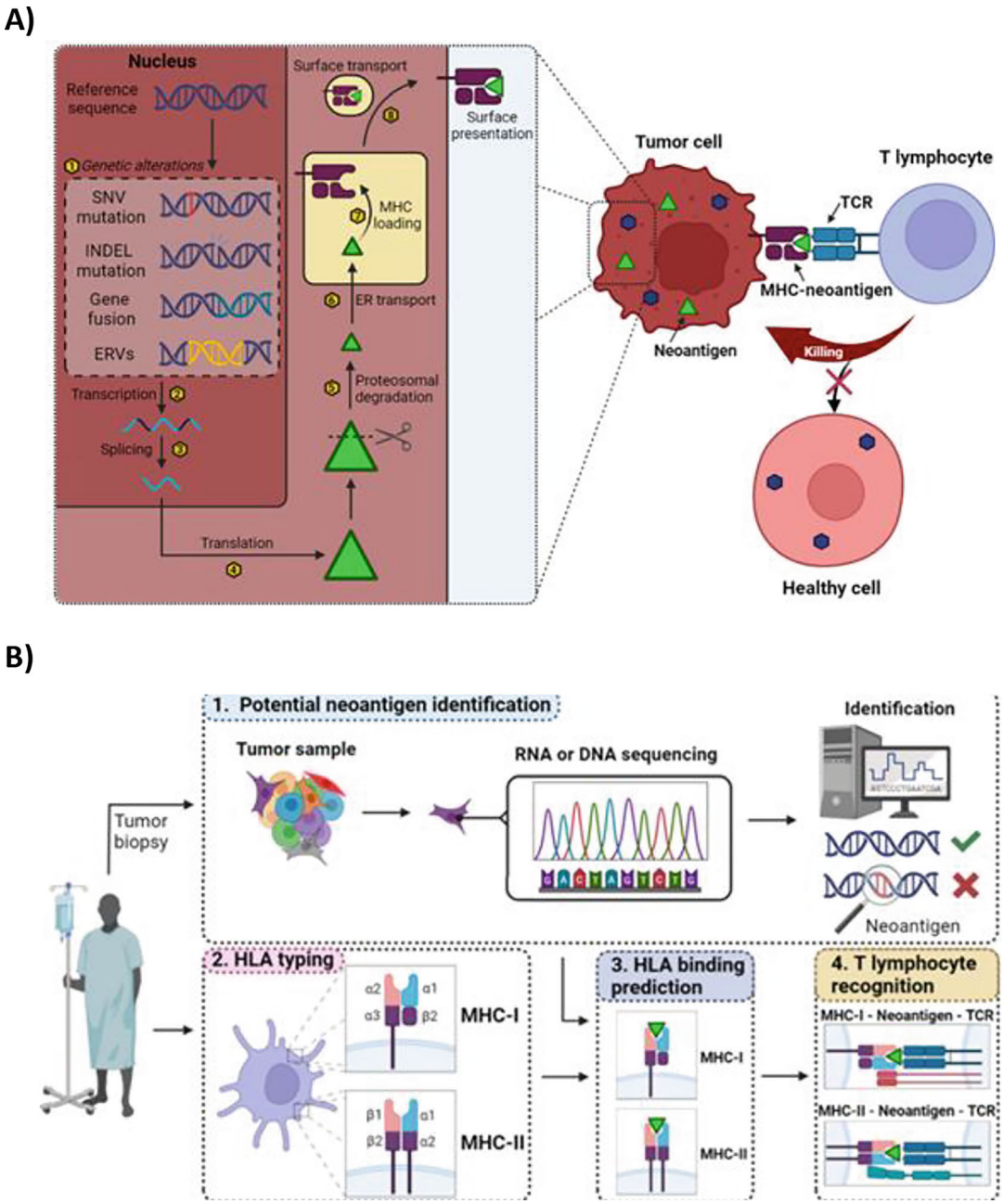


Fig. 1 (**a**) Key steps of neoantigens origin and processing until loaded in MHC molecules for their recognition by T cells: mutations can occur at a genomic level of the malignant cell (1), where they are transcribed (2) and spliced to form mRNA (3). During this process, alternative splicing can also produce splice variant mRNA. Translation of these variant mRNAs then leads to the synthesis of variant proteins (4). At this stage, post-transcriptional frameshifts, such as those caused by ribosomal slippage, can also produce variant proteins. These variant proteins can then undergo proteasomal degradation (5) and be transported to the endoplasmic reticulum (ER) (6), where they are subsequently loaded onto major histocompatibility complexes (MHCs) (7). After being loaded, the resulting neoantigen-MHC complexes can be transported to the cell surface (8), where they are exposed to recognition by the T-cell receptor (TCR) of lymphocytes. (**b**) Schematic representation of the overall process of identification of new neoantigens: Tumor samples are obtained and used to identify

Thus, the prediction of neoantigens is a critical process in the pursuit of truly personalized cancer immunotherapies, relying on advanced bioinformatics tools to integrate high-quality patient data with a rapidly expanding body of immunological knowledge. The overarching goal is to predict if patient-specific cancerous mutations can stimulate the immune system to target and eliminate the patient's own tumor.

In Subheading 2, an overview of the key steps in neoantigen prediction using NGS will be provided. Subheading 3 shows an example of neoantigenicity prediction comparing different methods over the same peptides. Afterwards, the available pipelines for neoantigen prediction and their characteristics will be listed in Subheading 4 for humans and in Subheading 5 for mouse models. Lastly, technical challenges and future improvements will be discussed in Subheading 6.

2 Key Steps of Neoantigen Prediction

The complex, multi-step process of neoantigen prediction involves several stages, each contributing to the final prediction. As explained before, the MHC Class I pathway is involved in presenting antigens originating from the inside of cells, for example, stemming from viruses and mutations, inducing CD8+ cytotoxic T cells. Traditionally, the MHC Class I antigen-presentation pathway has been recognized as the most restrictive, and consequently, the most predictive pathway for neoantigen prediction [10] and will thus be the main focus of this section. The key parts of neoantigen prediction process can be split into the following steps (Fig. 2):

2.1 Sample Collection

This is the initial, and arguably the most important, step in the process since all downstream results inadvertently depend on it. It involves obtaining high-quality patient samples from both tumor and representative normal tissues and comparing them to identify unique genetic alterations in the cancer cells that are not present in normal (germline) cells [11, 12]. These somatic mutations can potentially give rise to neoantigens, forming the basis for all downstream investigations. However, the mutations must be identified

Fig. 1 (continued) individualized neoantigens via RNA sequencing (RNAseq) or whole genome/exome sequencing (1). In parallel, HLA typing prediction is performed (2). Then a neoantigen-MHC complex binding prediction is evaluated (3). Once a suitable neoantigen-MHC complex is identified, the T-cell recognition via TCR is evaluated to check that a proper immune response can be triggered (4). In that case, the neoantigen discovered is classified as useful for therapeutic approaches. *SNV* single nucleotide variant, *INDEL* insertion/deletion, *ERV* endogenous retroviruses, *DNA* deoxyribonucleic acid, *MHC* major histocompatibility complex, *ER* endoplasmic reticulum, *TCR* T-cell receptor, *RNA* ribonucleic acid, *HLA* human leukocyte antigen. (Images created with BioRender)

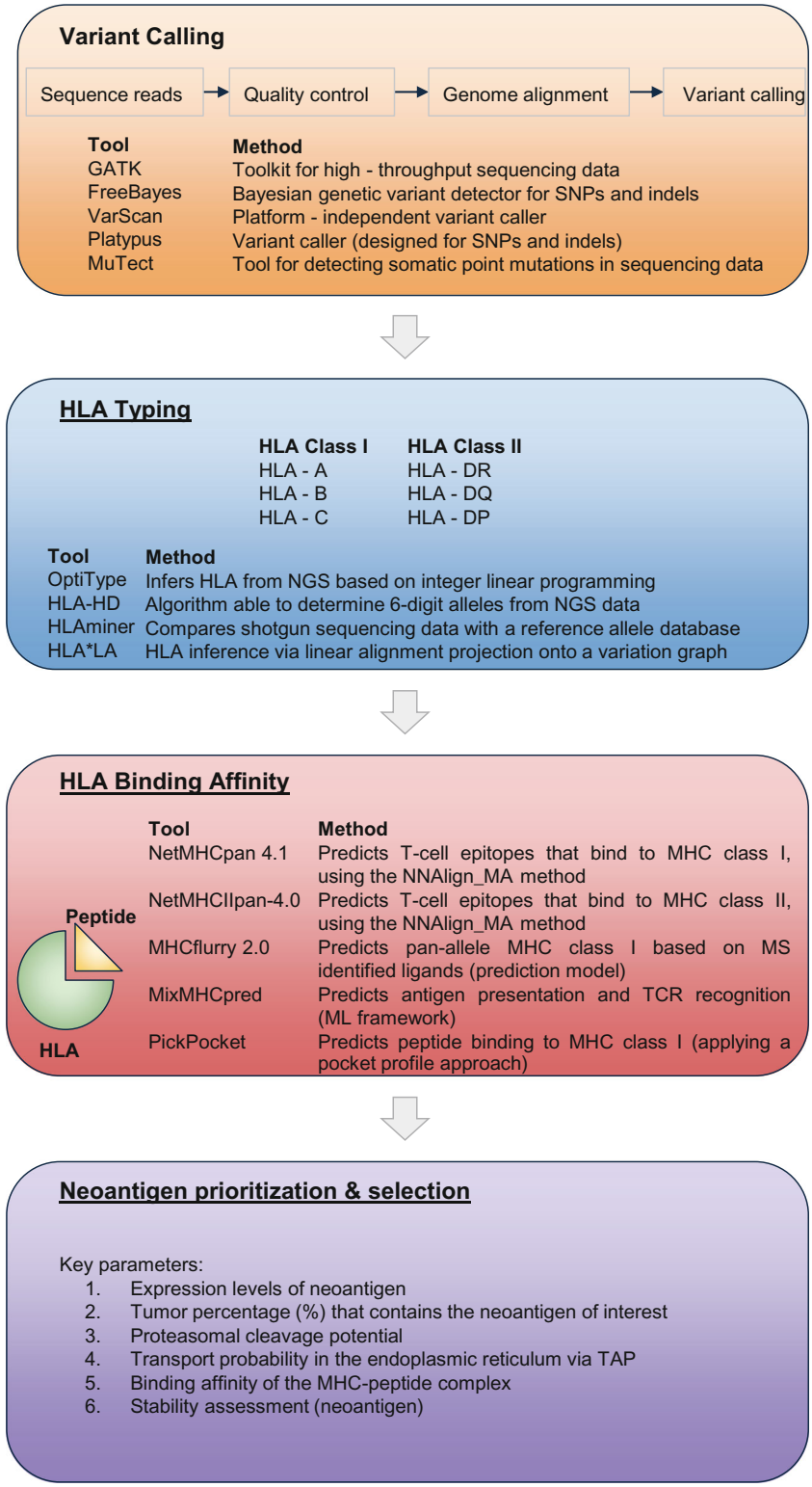


Fig. 2 Representation of a typical computational workflow integrating all the steps required for neoantigen prediction. Step 1 corresponds to variant calling where input data are processed to identify variants and annotate tumor-specific mutations. Step 2 refers to HLA typing where both Class I and Class II HLA alleles of

first, which requires the patient samples to be collected and stored in a manner that preserves the integrity of their genetic material.

2.2 Sequencing

These high-quality normal and tumor tissue samples are then typically sequenced using NGS techniques such as whole exome sequencing (WES) or whole genome sequencing (WGS). NGS can also be used for sequencing RNA, providing information on the expression level of specific somatic mutations [13].

2.3 Somatic Variant Calling

The sequenced reads from normal and tumor tissues are aligned to a reference genome, and the differences between them are identified. Specialized bioinformatics tools are used to perform this variant calling [14], and the identified somatic mutations are annotated and stored in variant call format (VCF) files, which contain essential details on each mutation. This includes information regarding nucleotide and amino acid changes, their genomic location, and additional annotations useful for downstream analyses and filtering, for example, variant allele frequency (VAF) scores, denoting the specific variant's prevalence.

2.4 In Silico Peptide Generation

Following the identification of patient-specific somatic variants, typically contained in VCF files, the next step is to construct all relevant peptides capable of containing these variants in silico. This involves using bioinformatics tools for transforming genomic nucleotide sequences into translated amino acid chains, constituting the proteins altered by the cancer. The subsequent splitting of proteins into mutation-containing peptides has no associated standard tool, but peptides of 8–12 amino-acid length, with a preference for 9-mers, are typically constructed for HLA Class I [15]. Longer peptides are constructed for HLA Class II, although the indicated range of relevant peptides can vary between studies, for example, 13–19 and 12–24 [16, 17].

2.5 Antigen Processing

The natural process emulated by in silico peptide generation occurs when proteins are damaged or decommissioned, and subsequently chopped up by proteasome enzymes into fragments. These peptide fragments are transported to the endoplasmic reticulum via transporter associated with antigen processing (TAP) proteins, where they can be further modified and finally loaded onto MHC molecules. This process is not random, however, and the uniformity

Fig. 2 (continued) patients are determined. Step 3 includes HLA binding affinity and stability prediction where in silico tools quantify the linkage among peptides-HLA alleles. Step 4 consists of neoantigen prioritization and selection aiming to facilitate the identification of immunogenic neoepitopes for personalized cancer immunotherapy. *SNP* single nucleotide polymorphism, *HLA* human leukocyte antigen, *NGS* next-generation sequencing, *MHC* major histocompatibility complex, *MS* mass spectrometry, *TCR* T-cell receptor, *ML* machine learning, *TAP* transporter associated with antigen processing

assumption of the in silico peptide generation needs modification. Computational immunology tools can help assess peptides, and their contextual flanking amino acids, by providing scores indicating the likelihood of, for example, cleavage sites and TAP compatibility [18, 19].

2.6 HLA Typing

The antigen-presentation pathway ultimately hinges on peptides binding to HLA molecules. Although in humans it is known as HLA, it is worth mentioning that the terms "MHC" and "HLA" are often used interchangeably. Thus, the patient's HLA alleles must be determined or typed [20]. There are three major HLA Class I genes: HLA-A, HLA-B, and HLA-C, with up to two alleles per gene, one from each biological parent. There are also several HLA Class II genes, which, unlike Class I, act as pairs forming heterodimer MHC molecules. HLA typing is performed by dedicated bioinformatics tools, providing allele identifiers denoting the amino acid composition of the resulting MHC molecules, and potentially also information affecting translation of the gene. The HLA genes are very polymorphic, and there are currently registered more than 27,000 Class I and 11,000 Class II allele variants ([21], v.3.56).

2.7 HLA Binding

The aim of HLA-binding prediction is estimating the binding affinity between all permutations of pairwise peptides and HLA alleles. This is a crucial step, since only peptides binding to HLA have a chance to be presented to the immune system. Machine learning, and neural networks in particular, have proven to be effective at this task, with well-known examples such as NetMHCpan [22] and MHCflurry [23]. These models are trained on large datasets of HLA-peptide pairs with associated experimentally determined binding affinities, typically given as half-maximal inhibitory concentration (IC50) values [24]. In the case of a competitive binding assay, an IC50 value represents the concentration of a candidate peptide required to outcompete a high-affinity reference peptide until only half of them remain bound to the HLA allele. There is a gradual transition from HLA-peptide pairs with strong binding affinities (small IC50 values) to weak binding affinities (large IC50 values), but a threshold of IC50 < 500 is often employed to denote Class I HLA-peptide pairs to be binders [25].

2.8 HLA-Peptide Stability

Similarly, measurements and predictions can be performed to assess the stability of HLA-peptide complexes. While closely related to HLA binding, stability revolves around measuring the HLA-peptide complex' ability to stay bound under perturbations, through, for example, thermal and kinetic stability assays [26]. Such stability assays work by subjecting the HLA-peptide complex to varying conditions and measuring how long it remains intact, providing valuable information about its durability. So,

while HLA binding assesses the potential of a peptide to bind to an HLA allele, HLA-peptide stability assesses the half-life of an HLA-peptide complex. A more stable complex might have a higher chance of being recognized by T cells, potentially leading to a stronger immune response [27].

In the context of the HLA-peptide complexes, the affinity/stability for the HLA is defined in its majority as the anchor residues of the peptides. These residues are buried within the HLA pocket and create bonds with the residues within it to stabilize the docking. In 9-mer peptides, the anchoring residues can normally be found at positions 2 and 9, although peptides of different lengths can have anchor residues in different positions [28].

2.9 Antigen Presentation

Somatic mutations capable of navigating through the above steps, forming stable HLA-peptide complexes, are then transported to the surface of the cell for subsequent inspection by the immune system. In essence, this is the first main goal of neoantigen prediction: predict the somatic mutations with the right characteristics to be transported as peptides to the cell surface. Mass spectrometry techniques can be used to discover these presented peptides, or eluted ligands, but cannot ascertain which HLA allele it was bound to [29]. It provides verification that a peptide indeed was presented. There are several ways this information could be used to increase predictive power, such as reinforcing HLA binding scoring [22, 30].

2.10 T-Cell Recognition and Immunogenicity

Once the HLA-peptide complexes are presented on the cell surface, they can be recognized by T cells, each carrying a unique T-cell receptor (TCR) capable of recognizing a few specific HLA-peptide complexes [31]. If a TCR binds to an HLA-peptide complex containing a somatic mutation, it triggers a signaling cascade that can initiate an immune response against these cells. This is the second main goal of neoantigen prediction: to predict which of the presented peptides will likely trigger an immune response. Predicting T-cell recognition is a huge challenge; however, the TCR diversity is vast, and the rules governing TCR-peptide-HLA interaction are not fully understood [31]. Some bioinformatics tools attempt to predict T-cell recognition by modeling TCR-peptide-MHC interactions [32], and while predictive power is steadily increasing, these predictions are often uncertain and require experimental validation [10, 33]. Laboratory experiments can be performed to validate that the neoantigens can elicit an immune response, with assays such as ELISA and intracellular cytokine staining [34].

2.11 Neoantigen Prioritization

Following the above steps, there are several predictions, scores, and potential immune-response measurements associated with each neoantigen candidate. Ranking and prioritizing these candidates by their overall potential for inducing an immune response is

challenging, and typically involves combining predictions of antigen processing, HLA binding and immunogenicity with gene expression and other relevant information into a single, combined score per neoantigen [10]. Recent advances within single-cell technologies and cutting-edge machine learning approaches are presumed to provide valuable insights for the future [35].

In conclusion, neoantigen prediction is a complex, multistep process that integrates numerous bioinformatics tools and data types in elaborated pipelines. There are several factors omitted in this section that could further increase the complexity, such as HLA Class II, posttranslational modifications, immune escape, mutated HLA genes, and alternative sequencing methods [7, 34, 36]. Despite the challenges, neoantigen prediction holds great promise for the transition toward truly personalized cancer immunotherapies.

3 Examples of Neoantigen Prediction Applying Different Tools to Cancer Driver Mutations

In order to illustrate the key steps described in the previous section and show how different tools perform on neoantigen prediction, an example using driver mutations will be developed in this section.

A variety of tools including those predicting binding affinity, stability, peptide cleavage, and TAP transportation will be tested. Although many tools focus on predicting HLA-binding affinity of mutated peptides, the two considered as gold standard will be used (NetMHCpan and MHCflurry, as discussed below) [22, 23]. The other tested tools focus on prediction of binding stability of the peptide-MHC complex (netMHCstabpan) [37], prediction of cleavage sites (netChop 3.1) [38], and TAP transport of the chopped peptides (TAP) [39]. Three well-known mutations have been selected: *KRAS* p.G12C, *TP53* p.H179R, and *GNAQ* p.Q209L. For the sake of clarity, only one allele (HLA-A*03–01) will be tested.

First, the mutated peptides (17-mers) need to be generated and written in FASTA format:

```
> sp|P01116|RASK_HUMAN - G12C
YKLVVVGACGVGKSALT

> sp|P04637|P53_HUMAN TP53 - H179R
EVVRRCPHRERCSDSDG

> sp|P50148|GNAQ_HUMAN Q209L
FRMVDVGGLRSERRKWI
```

3.1 Predicting Affinity and Stability

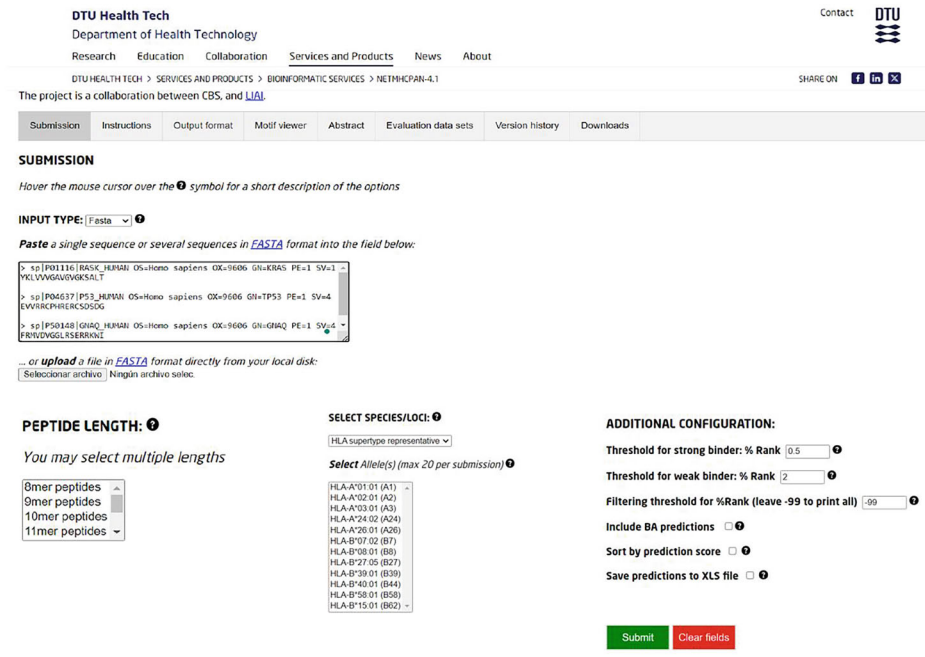
Many deep learning-based tools have been developed over the last years, but they all share the main functionality of predicting the HLA-binding affinity.

A work by Zhao et al. tested 18 antigen prediction tools and compared their capacity to accurately find binders by testing peptides of different lengths in a variety of HLAs [40]. In summary, MHCflurry and NetMHCpan outperformed other tools in peptide-MHC-binding prediction. Both tools are based on artificial neural networks (ANN) and exhibited superior performance in binary classification compared to logistic regression-based methods. This is because ANN-based approaches are favored for their ability to capture complex interactions between MHC-binding residues and for better regularization, reducing overfitting. MHCflurry consistently demonstrated robust performance in binder prediction. Moreover, MHCflurry was superior in three-class classification, suggesting its efficacy not only in identifying MHC binders but also in distinguishing strong binders. Nevertheless, one of the downsides of using this type of tools is the available data used for their training. The performance of AI-based tools largely depends on the quality and comprehensiveness of their training data. These tools use both quantitative (binding affinity) and qualitative (eluted ligand) assays. The latter yield binary outcomes, indicating only whether a peptide binds or not, without quantifying the strength of that binding in traditional affinity units such as nM. This poses a challenge for prediction tools like MHCflurry and NetMHCpan, which predict affinity values. The researchers who developed the tools came up with different strategies to assign affinity values to data lacking precise measurements, but these estimates may not always align with reality. Despite these shortcomings, NetMHCpan 4.1 and MHCflurry are still considered the gold standard in the field. Both tools can predict the affinity between the peptides and HLA allele input, among other neoantigen presentation related scores. They also have a very useful feature which is the splitting of protein sequences in all the possible peptides of a specified length. For these reasons, the examples below will focus on these tools.

3.1.1 NetMHCpan 4.1

NetMHCpan is one of the most well-known tools for MHC affinity predictions. It allows the user to predict the affinity of peptides for the different existing HLAs in different scores. In netMHCpan, the important information to define a binder or not, and to distinguish between strong and weak binders, comes from the predicted binding affinity and eluted ligand score. Apart from the raw score NetMHCpan provides %Rank scores, which indicate whether a predicted binding score stands relative to a set of random natural peptides. Unlike some scoring methods that can be skewed by the tendency of certain molecules to generally have higher or lower predicted affinities, the rank values are designed to be bias-free,

A)



B)

```
HLA-A03:01 : Distance to training data  0.000 (using nearest neighbor HLA-A03:01)

# Rank Threshold for Strong binding peptides   0.500
# Rank Threshold for Weak binding peptides   2.000
-----------------------------------------------------------------------------------------------------------------------
 Pos         MHC        Peptide      Core Of Gp Gl Ip Il        Icore        Identity  Score_EL %Rank_EL Score_BA %Rank_BA  Aff(nM) BindLevel
-----------------------------------------------------------------------------------------------------------------------
   5 HLA-A*03:01      VVGACGVGK VVGACGVGK  0  0  0  0  0    VVGACGVGK sp_P01116_RASK_ 0.2949510    0.871 0.503725    0.671   214.77 <= WB
   7 HLA-A*03:01      GACGVGKSA GACGVGKSA  0  0  0  0  0    GACGVGKSA sp_P01116_RASK_ 0.0000590   41.333 0.023968   67.982 38578.48
   3 HLA-A*03:01      LVVVGACGV LVVVGACGV  0  0  0  0  0    LVVVGACGV sp_P01116_RASK_ 0.0000440   46.000 0.057037   27.701 26974.59
   2 HLA-A*03:01      KLVVVGACG KLVVVGACG  0  0  0  0  0    KLVVVGACG sp_P01116_RASK_ 0.0000390   48.000 0.057684   27.301 26786.42
   8 HLA-A*03:01      ACGVGKSAL ACGVGKSAL  0  0  0  0  0    ACGVGKSAL sp_P01116_RASK_ 0.0000240   56.429 0.016001   84.254 42051.54
   9 HLA-A*03:01      CGVGKSALT CGVGKSALT  0  0  0  0  0    CGVGKSALT sp_P01116_RASK_ 0.0000080   75.000 0.026008   64.019 37736.29
   4 HLA-A*03:01      VVVGACGVG VVVGACGVG  0  0  0  0  0    VVVGACGVG sp_P01116_RASK_ 0.0000060   80.000 0.039669   43.098 32551.18
   6 HLA-A*03:01      VGACGVGKS VGACGVGKS  0  0  0  0  0    VGACGVGKS sp_P01116_RASK_ 0.0000050   82.500 0.023438   69.033 38800.35
   1 HLA-A*03:01      YKLVVVGAC YKLVVVGAC  0  0  0  0  0    YKLVVVGAC sp_P01116_RASK_ 0.0000030   87.500 0.012605   90.642 43625.40
-----------------------------------------------------------------------------------------------------------------------

Protein sp_P01116_RASK_. Allele HLA-A*03:01. Number of high binders 0. Number of weak binders 1. Number of peptides 9
```

C)

```
Link to Allele Frequencies in Worldwide Populations HLA-A03:01
-----------------------------------------------------------------------------

# Rank Threshold for Strong binding peptides   0.500
# Rank Threshold for Weak binding peptides   2.000
-----------------------------------------------------------------------------------------------------------------------
 Pos         MHC        Peptide      Core Of Gp Gl Ip Il        Icore        Identity  Score_EL %Rank_EL Score_BA %Rank_BA  Aff(nM) BindLevel
-----------------------------------------------------------------------------------------------------------------------
   1 HLA-A*03:01      EVVRRCPHR EVVRRCPHR  0  0  0  0  0    EVVRRCPHR sp_P04637_P53_H 0.0043080    7.830 0.105705   11.974 15931.82
   3 HLA-A*03:01      VRRCPHRER VRRCPHRER  0  0  0  0  0    VRRCPHRER sp_P04637_P53_H 0.0012950   12.585 0.060651   25.541 25940.17
   2 HLA-A*03:01      VVRRCPHRE VVRRCPHRE  0  0  0  0  0    VVRRCPHRE sp_P04637_P53_H 0.0008770   14.665 0.074188   19.439 22405.92
   4 HLA-A*03:01      RRCPHRERC RRCPHRERC  0  0  0  0  0    RRCPHRERC sp_P04637_P53_H 0.0001170   32.000 0.022652   70.604 39131.71
   5 HLA-A*03:01      RCPHRERCS RCPHRERCS  0  0  0  0  0    RCPHRERCS sp_P04637_P53_H 0.0000030   87.500 0.009742   95.093 44997.93
   6 HLA-A*03:01      CPHRERCSD CPHRERCSD  0  0  0  0  0    CPHRERCSD sp_P04637_P53_H 0.0000020   90.000 0.015796   84.670 42144.90
   9 HLA-A*03:01      RERCSDSDG RERCSDSDG  0  0  0  0  0    RERCSDSDG sp_P04637_P53_H 0.0000010   95.000 0.014771   86.637 42614.91
   8 HLA-A*03:01      HRERCSDSD HRERCSDSD  0  0  0  0  0    HRERCSDSD sp_P04637_P53_H 0.0000000  100.000 0.006069   96.943 46822.21
   7 HLA-A*03:01      PHRERCSDS PHRERCSDS  0  0  0  0  0    PHRERCSDS sp_P04637_P53_H 0.0000000  100.000 0.007033   96.457 46336.39
-----------------------------------------------------------------------------------------------------------------------

Protein sp_P04637_P53_H. Allele HLA-A*03:01. Number of high binders 0. Number of weak binders 0. Number of peptides 9
```

D)

```
Link to Allele Frequencies in Worldwide Populations HLA-A03:01
-----------------------------------------------------------------------------

# Rank Threshold for Strong binding peptides   0.500
# Rank Threshold for Weak binding peptides   2.000
-----------------------------------------------------------------------------------------------------------------------
 Pos         MHC        Peptide      Core Of Gp Gl Ip Il        Icore        Identity  Score_EL %Rank_EL Score_BA %Rank_BA  Aff(nM) BindLevel
-----------------------------------------------------------------------------------------------------------------------
   2 HLA-A*03:01      RMVDVGGLR RMVDVGGLR  0  0  0  0  0    RMVDVGGLR sp_P50148_GNAQ_ 0.2605480    0.973 0.430447    1.128   474.58 <= WB
   7 HLA-A*03:01      GGLRSERRK GGLRSERRK  0  0  0  0  0    GGLRSERRK sp_P50148_GNAQ_ 0.0129680    4.960 0.130485    8.905 12184.98
   8 HLA-A*03:01      GLRSERRKW GLRSERRKW  0  0  0  0  0    GLRSERRKW sp_P50148_GNAQ_ 0.0017940   11.083 0.051454   31.580 28654.26
   5 HLA-A*03:01      DVGGLRSER DVGGLRSER  0  0  0  0  0    DVGGLRSER sp_P50148_GNAQ_ 0.0017260   11.264 0.055181   28.868 27521.75
   6 HLA-A*03:01      VGGLRSERR VGGLRSERR  0  0  0  0  0    VGGLRSERR sp_P50148_GNAQ_ 0.0002510   23.862 0.041453   41.045 31928.87
   3 HLA-A*03:01      MVDVGGLRS MVDVGGLRS  0  0  0  0  0    MVDVGGLRS sp_P50148_GNAQ_ 0.0002340   24.520 0.057349   27.508 26883.68
   1 HLA-A*03:01      FRMVDVGGL FRMVDVGGL  0  0  0  0  0    FRMVDVGGL sp_P50148_GNAQ_ 0.0000290   53.125 0.027184   61.868 37259.17
   4 HLA-A*03:01      VDVGGLRSE VDVGGLRSE  0  0  0  0  0    VDVGGLRSE sp_P50148_GNAQ_ 0.0000230   57.143 0.013562   88.933 43176.00
   9 HLA-A*03:01      LRSERRKWI LRSERRKWI  0  0  0  0  0    LRSERRKWI sp_P50148_GNAQ_ 0.0000120   68.750 0.018759   78.617 40815.20
-----------------------------------------------------------------------------------------------------------------------

Protein sp_P50148_GNAQ_. Allele HLA-A*03:01. Number of high binders 0. Number of weak binders 1. Number of peptides 9
```

Fig. 3 (**a**) User interface for the NetMHCpan tool for MHC affinity prediction. Input box to enter the full protein sequence containing the neoantigen of interest as FASTA, or directly input the peptide, to select peptide length and HLA alleles, as well as other filters and optional settings for the output. (**b**–**d**) Predictions from mutated

ensuring an accurate comparison across different peptides. If the %Rank score for either binding affinity or eluted ligand is above 2.0, the peptide is considered a non-binder. If the score falls between 2.0 and 0.5 (exclusive), the peptide is classified as a weak binder. If the score is equal to or under 0.5, the peptide is labeled as a strong binder. NetMHCpan also offers a column called "Binding level," which points to the binding strength based on the later values, classifying peptides as weak binder (WB), and strong binder (SB). The evaluation of peptide binding to MHC molecules is often assessed through various scoring systems such as the IC50. The IC50 value represents the peptide's binding affinity to MHC, with lower values indicating stronger binding. For instance, an IC50 of 500 nM or 50 nM typically denotes peptide binders or strong binders to MHC, respectively [22].

NetMHCpan can be used through the interface available or the command line version. The interface version is shown in Fig. 3a. NetMHCpan outputs the result in tables containing information of all the evaluated 9-mer peptides, their affinity, and their ranks both for binding affinity and for eluted ligand, as well as a final column indicating whether the peptide is a strong or weak binder. Figure 3b–d shows the resulting predictions of the mutated peptides used for this example. Evaluation of the selected 17-mers resulted in two potential neoantigens. The 9-mer peptides were RMVDVGG**L**R, VVGA**C**GVGK, from *GNAQ* p.Q209L and *KRAS* p.G12C, respectively. These peptides passed both %Rank (BA and EL) scores as Weak Binders (under 2.0, but above 0.5). Since the different tools might have discrepancies, the same peptides were evaluated using MHCflurry to confirm their neoantigenicity.

3.1.2 *MHCflurry 2.0*

MHCflurry implements Class I peptide/MHC-binding affinity prediction. MHCflurry also includes two experimental predictors: an "antigen processing" predictor that attempts to model MHC allele-independent effects such as proteasomal cleavage and a "presentation" predictor that integrates processing predictions with binding affinity predictions to give a composite "presentation score." Both models are trained using binding affinity and mass spectrometry eluted ligand assays [23].

MHCflurry has the same functionality as netMHCpan, but runs in Python. The input can be a full protein sequence, which is

Fig. 3 (continued) peptides by NetMHCpan for (**b**) *KRAS* p.G12C, (**c**) *TP53* p.H179R, and (**d**) *GNAQ* p.Q209L mutations. Pos, indicates the residue number of the peptide in the protein sequence, starting from 0; MHC, specifies the HLA allele or supertipe; Score_EL, is the raw prediction score for eluted ligand; %Rank_EL, is the rank of the predicted binding score compared to a set of random natural peptides; Aff, affinity; BindLevel, indicates the binding level, where SB stands for strong binder and WB for weak binder; *HLA* human leukocyte antigen, *MHC* major histocompatibility complex, *nM* nanoMolar

A)

pos	peptide	n_flank	c_flank	sample_name	*affinity*	best_allele	affinity_percentile	processing_score	presentation_score	presentation_percentile
4	VVGACGVGK	YKLV	SALT	sample1	88.1087177956	A0301	0.356625	0.2948311158	0.7180030017	0.407826087
2	LVVVGACGV	YK	GKSAL	sample1	24764.5673555715	A0301	11.878	0.0460864494	0.0040635644	62.744673913
6	GACGVGKSA	KLVVV	LT	sample1	27047.3267352356	A0301	15.252375	0.0641564511	0.0039947597	99.2866032609
3	VVVGACGVG	YKL	KSALT	sample1	27446.6044791236	A0301	15.813875	0.0424762408	0.0036278675	99.2866032609
1	KLVVVGACG	Y	VGKSA	sample1	27516.1414633246	A0301	15.813875	0.0911028546	0.0043507799	62.744673913
7	ACGVGKSAL	LVVVG	T	sample1	27768.0887766753	A0301	16.440875	0.0409491413	0.0035664157	99.2866032609
5	VGACGVGKS	YKLVV	ALT	sample1	30137.2671019385	A0301	24.726625	0.0064701785	0.0028904508	100
8	CGVGKSALT	VVVGA		sample1	30978.0759969185	A0301	28.558125	0.0559852204	0.0033952934	99.2866032609
0	YKLVVVGAC		GVGKS	sample1	33569.7738385478	A0301	71.81975	0.1942298999	0.0053010713	46.224673913

B)

pos	peptide	n_flank	c_flank	sample_name	*affinity*	best_allele	affinity_percentile	processing_score	presentation_score	presentation_percentile
0	EVVRRCPHR		ERCSD	sample1	5744.16002...	A0301	2.862	0.4129955769	0.0639445992	4.1571195652
2	VRRCPHRFR	EV	CSDSD	sample1	17211.5580...	A0301	6.47125	0.0830098669	0.0066467655	37.359048913
1	VVRRCPHRE	E	RCSDS	sample1	23378.3835...	A0301	10.26625	0.019987814	0.0038926161	99.2866032609
3	RRCPHRERC	EVV	SDSDG	sample1	30418.5527...	A0301	26.48575	0.0106666694	0.0029104374	100
4	RCPHRERCS	EVVR	DSDG	sample1	31189.8780...	A0301	31.022125	0.0004339218	0.0027324261	100
8	RERCSDSDG	RRCPH		sample1	32154.5158...	A0301	41.914375	0.0048288169	0.0026973716	100
5	CPHRERCSD	EVVRR	SDG	sample1	32232.4733...	A0301	41.914375	0.0005860818	0.0026480847	100
7	HRERCSDSD	VRRCP	G	sample1	32570.1713...	A0301	47.31825	0.0001834681	0.0026174171	100
6	PHRERCSDS	VVRRC	DG	sample1	33219.3063...	A0301	62.27625	0.0009916644	0.0025756128	100

C)

pos	peptide	n_flank	c_flank	sample_name	*affinity*	best_allele	affinity_percentile	processing_score	presentation_score	presentation_percentile
1	RMVDVGGLR	F	SERRK	sample1	54.6607075396	A0301	0.1915	0.5723558851	0.920906192	0.0904347826
6	GGLRSERRK	RMVDV	WI	sample1	6936.977180565	A0301	3.153625	0.002089716	0.0117762514	20.2322554348
4	DVGGLRSER	FRMV	RKWI	sample1	7467.0617919316	A0301	3.301	0.9150099531	0.2630452771	1.5116032609
5	VGGLRSERR	FRMVD	KWI	sample1	18239.7721678958	A0301	6.912875	0.0085001605	0.0047409168	62.744673913
7	GLRSERRKW	MVDVG	I	sample1	23208.061240231	A0301	10.26625	0.0522665373	0.00443	62.744673913
3	VDVGGLRSE	FRM	RRKWI	sample1	27228.3615168625	A0301	15.252375	0.4686669018	0.0182147959	12.8054619565
2	MVDVGGLRS	FR	FRRKW	sample1	27577.4211648356	A0301	16.440875	0.2779251873	0.0087942378	27.6173913043
0	FRMVDVGGL		RSERR	sample1	29585.2449192134	A0301	21.8915	0.4013217771	0.0130729844	17.3055978261
8	LRSERRKWI	VDVGG		sample1	31189.2817495858	A0301	31.022125	0.0810565602	0.0037091077	99.2866032609

Fig. 4 Predictions from mutated peptides by MHCflurry for peptides containing (**a**) *KRAS* p.G12C mutation, (**b**) *TP53* p.H179R, and (**c**) *GNAQ* p.Q209L mutations. Pos, indicates the residue number of the peptide in the protein sequence, starting from 0

splitted into all the possible peptides of the specified length (9-mers in this example) to predict the binding affinities and calculate the scores of each resulting peptide. The sequences have to be input as a Python dictionary using the argument "sequences." Then, the argument "alleles" takes a list with the HLA alleles that are going to be used for the predictions. The length of the peptides is called "peptide_lengths," which takes a list with the desired input lengths. An example of how to run MHCflurry is shown below:

```
from mhcflurry import Class1PresentationPredictor
predictor = Class1PresentationPredictor.load()

predictor.predict_sequences(
```

```
sequences={'RASK_HUMAN - G12C': "YKLVVVGACGVGKSALT",
'P53_HUMAN TP53 - H179R': "EVVRRCPHRERCSDSDG",
'GNAQ_HUMAN Q209L': "FRMVDVGGLRSERRKWI"},

alleles={"sample1": ["A0301"] },
result="all",
peptide_lengths= [9],
throw = False,
verbose=0)
```

In this case, it does not calculate %Rank like NetMHCpan but outputs a percentile, which can be used to define a binder or not. Any peptide with an affinity and/or presentation percentiles that falls between 0 and 5 can be considered a binder or a presented peptide. However, this is not a validated range. The well-validated threshold is affinity of 500 nM. Figure 4 shows the resulting predictions of the mutated peptides used for this example. The two peptides classified as HLA-A*03–01 binders with NetMHCpan were also classified as binders by MHCflurry, showing concordance between these two tools.

In summary, only one out of the 9-mers generated from *KRAS* p.G12C mutation has been identified as a potential neoantigen, although classified as weak. Also, one 9-mer from the *GNAQ* p.Q209L mutation has been identified as a potential neoantigen. In the case of *TP53* p.H179R, none of the generated 9-mers had affinity for the tested HLA. Therefore, it will be excluded from downstream analyses. Next, the stability of the peptide-MHC complex in *KRAS* p.G12C and *GNAQ* p.Q209L putative neoantigens will be evaluated.

3.1.3 NetMHCstabpan 1.0

NetMHCstabpan is a computational tool designed to predict the stability of peptide-MHC Class I (pMHC-I) complexes. It uses a machine learning approach to forecast the half-life of peptides bound to MHC molecules, an essential step in antigen presentation by the immune system. It is possible to run this tool using the interface or the command line version [37] (Fig. 5a).

As in netMHCpan, the %rank values are not affected by the inherent bias of certain molecules toward higher or lower mean predicted affinities. Strong binders are defined as having %rank <0.5, and weak binders with %rank <2. Of note, this tool also includes a combined score using the affinity and stability values. In fact, the output information is similar to netMHCpan, except for the two new parameters related to stability and combined scores of affinity and stability (Fig. 5b, c).

As a result, the two peptides from *KRAS* p.G12C and *GNAQ* p.Q209L mutations, RMVDVGGLR and VVGACGVGK, were predicted as stable binders. Thus, netMHCstabpan confirmed

A)

B)

HLA-A03:01 : Distance to traning data 0.000 (using nearest neighbor HLA-A03:01)

Rank Threshold for Strong binding peptides 0.500
Rank Threshold for Weak binding peptides 2.000

pos	HLA	peptide	Identity	Pred	Thalf(h)	%Rank_Stab	1-log50k	Aff(nM)	%Rank_aff	Combined	Combined_%rank	BindLevel
4	HLA-A*03:01	VVGACGVGK	sp_P01116_RASK_	0.488	0.97	1.10	0.491	246.31	1.00	0.490	1.02	<= WB
1	HLA-A*03:01	KLVVVGACG	sp_P01116_RASK_	0.023	0.18	16.00	0.051	28777.30	32.00	0.045	26.67	
2	HLA-A*03:01	LVVVGACGV	sp_P01116_RASK_	0.004	0.12	45.00	0.045	30846.65	50.00	0.037	48.91	
6	HLA-A*03:01	GACGVGKSA	sp_P01116_RASK_	0.013	0.16	24.00	0.022	39587.82	50.00	0.020	41.10	
3	HLA-A*03:01	VVVGACGVG	sp_P01116_RASK_	0.003	0.12	55.00	0.023	39143.57	50.00	0.019	50.93	
5	HLA-A*03:01	VGACGVGKS	sp_P01116_RASK_	0.001	0.10	80.00	0.022	39217.35	50.00	0.018	54.05	
8	HLA-A*03:01	CGVGKSALT	sp_P01116_RASK_	0.001	0.10	80.00	0.019	40738.44	50.00	0.015	54.05	
7	HLA-A*03:01	ACGVGKSAL	sp_P01116_RASK_	0.001	0.10	80.00	0.013	43405.52	50.00	0.011	54.05	
0	HLA-A*03:01	YKLVVVGAC	sp_P01116_RASK_	0.001	0.09	80.00	0.008	45816.91	50.00	0.007	54.05	

Protein sp_P01116_RASK_. Allele HLA-A*03:01. Number of high binders 0. Number of weak binders 1. Number of peptides 9

C)

Link to Allele Frequencies in Worldwide Populations HLA-A*03:01

Rank Threshold for Strong binding peptides 0.500
Rank Threshold for Weak binding peptides 2.000

pos	HLA	peptide	Identity	Pred	Thalf(h)	%Rank_Stab	1-log50k	Aff(nM)	%Rank_aff	Combined	Combined_%rank	BindLevel
1	HLA-A*03:01	RMVDVGGLR	sp_P50148_GNAQ_	0.477	0.94	1.10	0.418	544.67	1.50	0.430	1.40	<= WB
6	HLA-A*03:01	GGLRSERRK	sp_P50148_GNAQ_	0.021	0.18	17.00	0.123	13174.60	15.00	0.103	15.36	
2	HLA-A*03:01	MVDVGGLRS	sp_P50148_GNAQ_	0.004	0.13	40.00	0.060	26246.48	32.00	0.049	33.33	
5	HLA-A*03:01	VGGLRSERR	sp_P50148_GNAQ_	0.002	0.11	55.00	0.050	29103.26	32.00	0.040	34.92	
7	HLA-A*03:01	GLRSERRKW	sp_P50148_GNAQ_	0.025	0.19	16.00	0.044	31170.41	50.00	0.040	35.09	
4	HLA-A*03:01	DVGGLRSER	sp_P50148_GNAQ_	0.013	0.16	24.00	0.042	31685.57	50.00	0.036	41.10	
0	HLA-A*03:01	FRMVDVGGL	sp_P50148_GNAQ_	0.001	0.09	80.00	0.016	42192.82	50.00	0.013	54.05	
8	HLA-A*03:01	LRSERRKWI	sp_P50148_GNAQ_	0.001	0.10	60.00	0.013	43458.15	50.00	0.011	51.72	
3	HLA-A*03:01	VDVGGLRSE	sp_P50148_GNAQ_	0.000	0.09	80.00	0.007	46451.85	50.00	0.006	54.05	

Protein sp_P50148_GNAQ_. Allele HLA-A*03:01. Number of high binders 0. Number of weak binders 1. Number of peptides 9

Fig. 5 (**a**) User interface for the NetMHCstabpan tool for prediction of the stability of peptide-MHC Class I complexes: Input box to enter the full protein sequence containing the neoantigen of interest as FASTA, or directly input the peptide, to select peptide length and HLA allele, as well as other filters and optional settings for the output. (**b**–**c**) Predictions from mutated peptides by netMHCstabpan for peptides containing

that those mutations could produce potential neoantigens, with affinity and stability for HLA-A*03:01. Nevertheless, additional steps are required to evaluate whether these peptides will be presented on the cell surface, such as prediction of cleavage sites.

3.2 Predicting Cleavage and Transport

3.2.1 NetChop 3.1

NetChop is a computational tool designed to predict cleavage sites for processing protein precursors into mature peptides by proteases within the MHC Class I antigen presentation pathway. It operates by using a neural network trained on a dataset of experimentally verified cleavage sites, learning patterns indicative of protease specificity. When provided with an amino acid sequence, NetChop assesses the likelihood of each residue being part of a cleavage site based on its surrounding sequence context and outputs a probability score for cleavage at each position [38].

NetChop also offers an interface closely similar to netMHCpan and netMHCstabpan (Fig. 6a). In this example, the prediction method was "C term 3.0," with the default threshold of "0.5" and output set to "Short output." NetChop gives as a result the input sequence marked with the predicted cleavage sites. The residue where the cleavage is most likely happening is marked with an "S," whereas, if the cleavage is not occurring, the residue is marked with a ".". If a residue is assigned with an "S" the peptide bond on the C-terminal side is cleaved.

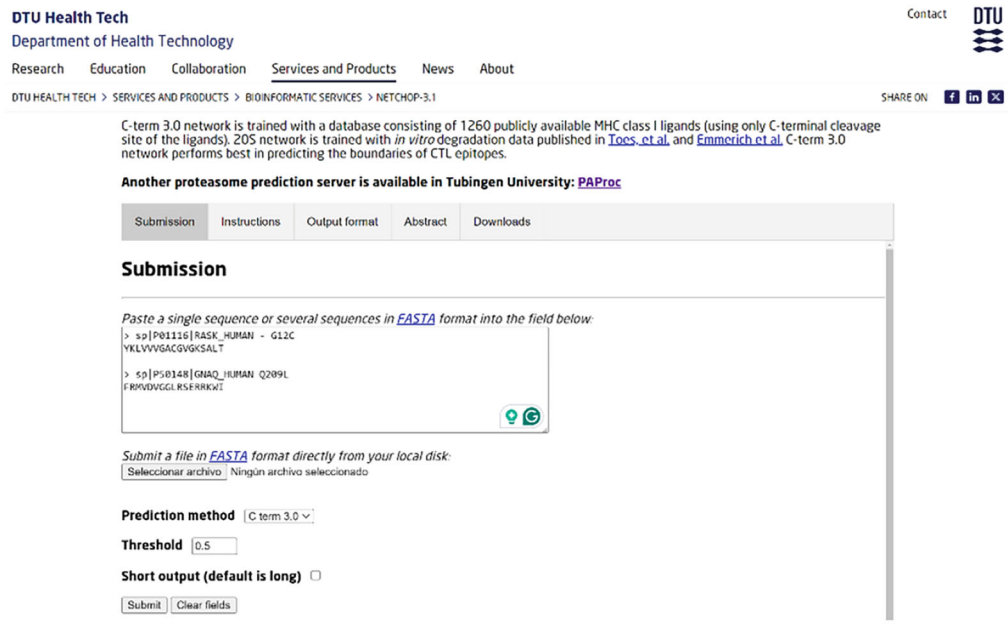
The cleavage sites are selected based on the value output by the tool. This value can go from 0 to 1 and everything above 0.5 is selected as a cleavage site. To access the predicted values per amino acid "Short output" option needs to be unselected.

Figure 6b shows the resulting cleavage predictions of the mutated peptides. In the case of *KRAS*, the 9-mer VVGACGVGK was not found within the array of peptides produced after chopping the mutated protein. On the contrary, the predicted cleavage pattern of the 17-mer *GNAQ* p.Q209L can produce the peptide RMVDVGGLR, which had binding affinity for HLA-A*03:01 allele (as shown above). It is important to note that the 9-mer RMVDVGGLR also contains internal cleavage sites, which could lead to the generation of shorter peptides.

Therefore, only the peptide RMVDVGGLR from *GNAQ* p.Q209L remains as a potential neoantigen and will be tested in the

Fig. 5 (continued) KRAS p.G12C and *GNAQ* p.Q209L mutations. Pos, indicates the residue number of the peptide in the protein sequence, starting from 0; HLA, specifies the MHC molecule or allele name; Pred, Stability prediction score; Thalf(h), The predicted half-life of the pMHC complex (in hours); %Rank_Stab, %Random - % Rank of predicted stability score to a set of 200,000 random natural 9-mer peptides; 1-log50K, Affinity Prediction score (called 1-log50K(aff)); Aff(nM), Affinity as IC50 value in nM (only for white-listed alleles); % Rank_aff, %Random - %Rank of predicted affinity score to a set of 200,000 random natural 9-mer peptides; Combined, Prediction score combining Affinity and Stability predictions; Combined_%rank, %Rank approximation using both stability and affinity %Rank; BindLevel, Binding level (*SB* strong binder, *WB* weak binder)

A)



B)

```
NetChop 3.0 predictions using version C-term. Threshold 0.500000

17  sp_P01116
YKLVVVGACGVGKSALT
S.S.SS.S..S.....
---------------------------------------

Number of cleavage sites 6. Number of amino acids 17. Protein name  sp_P01116

---------------------------------------
17  sp_P50148
FRMVDVGGLRSERRKWI
S..S....SS.....SS
---------------------------------------

Number of cleavage sites 6. Number of amino acids 17. Protein name  sp_P50148
```

Fig. 6 (**a**) User interface for the netChop 3.1 tool for prediction of the peptide cleavage sites: Input box to enter the full protein sequence containing the neoantigen of interest as FASTA, or input the peptide directly, together with optional settings. (**b**) Predictions of cleavage sites by netChop 3.1 for the 17-mer containing the *GNAQ* p.Q209L mutation

last step, intending to predict if this peptide could be presented on the cell surface of cancer cells by being transported to the ER by TAP protein.

3.2.2 TAP Transport Predictions

For the TAP transport prediction, many AI-based tools have recently been developed. A validated method described by Peters et al. will be used in this example. The scoring matrix and how the values are assigned for every amino acid at each position of a 9-mer (calculated from experimental data) and formula used to score each 9-mer can be found in the original work [39].

The TAP score using this method on RMVDVGGLR was −1.97. Since the authors specified that any TAP score below 1 would be considered a potential neoantigen, this peptide would pass all thresholds to be considered as a potential neoantigen.

In summary, according to our findings; from the three evaluated mutations, only *GNAQ* p.Q209L mutation would have a potential neoantigen, in agreement with previous reports. Interestingly, *GNAQ* p.Q209L mutation is harboured by approximately 70% patients with uveal melanoma so it could represent a therapeutic opportunity [41, 42]. In contrast, none of the peptides resulting from *TP53* p.H179R mutation had sufficient affinity to bind HLA so would not be immunogenic, in agreement with previous studies [42]. Whilst other mutations arising from *TP53* could lead to potential neoantigens [43], it has been reported that *TP53* mutations with greater oncogenic potential would be less immunogenic [44]. Finally, although one peptide derived from *KRAS* p.G12C mutation scored as a potential HLA-binding peptide, this 9-mer would not be generated according to the cleavage prediction tool. Of note, more potential neoantigens would have been found if more HLA genotypes had been interrogated. In that sense, the fact that neoantigens are restricted to certain HLA alleles limits the applicability for immunotherapeutic approaches [45, 46].

4 Overview of Available Computational End-to-End Workflows for Neoantigen Identification

Computational workflows for neoantigen prediction consist of four main steps, which can be categorized as (1) variant calling and annotation for tumor-specific mutations, (2) HLA genotyping of patients' alleles, (3) prediction of HLA-binding affinity and stability of peptide epitopes, and (4) neoantigen prioritization and identification of immunogenic neoepitopes for personalized cancer immunotherapy (Fig. 2). Those steps, explained in Subheading 2, are typically integrated into a sophisticated bioinformatics pipeline provided as a ready-to-use software. With the advent of AI, some of the steps are being replaced or facilitated by machine learning algorithms, whereas recent state-of-the-art approaches utilize end-to-end deep learning models. Over the past years, several methods have been proposed to conduct each step of the neoantigen prediction process. As an example, in Subheading 3 an extensive

description of the available methods for HLA-binding affinity and stability prediction has been provided. Herein, an overview of the methods providing unified pipelines and workflows for neoantigen prediction will be presented. Since several end-to-end pipelines are available, this section will focus on those that provide unique features to users aimed at improving neoantigen identification, facilitating clinical implementation, reducing computation time, and inclusiveness in the user group.

CloudNeo [47]: It was the initial effort to introduce a cloud-based neoantigen prediction workflow with the scope of identifying patient-specific tumor neoantigens. The CloudNeo workflow requires non-synonymous mutations in VCF format and RNA or DNA sequencing data in BAM format for HLA typing. Then, the VEP tool [48] and a custom R script named Protein_Translator are utilized to convert genomic variants into amino acid changes. The Protein_Translator generates a list of N-amino-acid-long peptide sequences in FASTA format, with the single peptide change positioned in the middle of the N-mer. Additionally, it generates another FASTA file for homologous N-mers without peptide mutations. Users can select either HLAminer [49] or Polysolver [50] to calculate six predicted HLA types (top two predictions each for HLA-A, HLA-B, and HLA-C). Then, the NetMHCpan tool [22, 51] computes binding affinities between the six HLA types and each ([N/2] + 1)-mer peptide subsequence within the N-mers. The platform's output includes peptide subsequences and MHC-binding affinity scores for all six HLA types. The CloudNeo pipeline, implemented in Common Workflow Language (CWL), is publicly available on GitHub. It can be executed using Rabix, enabling deployment on various platforms such as AWS, Google Compute Engine, and Azure.

Antigen.garnish [52] is a workflow featuring unique characteristics. The first is ensemble neoantigen prediction, while the second is utilizing the dissimilarity to the non-mutated (reference) proteome to identify high-quality predicted neoantigens. More specifically, the antigen.garnish workflow input options include VCFs, peptide sequences, or ensemble transcript IDs with HGVS-style cDNA annotations. Mutated sequences undergo prediction and filtering against the non-mutated proteome. The main function of the tool, "garnish_affinity," is to conduct ensemble MHC affinity prediction. The ensemble method generates a unified affinity score by averaging the affinities from all models predicting the peptide-MHC binding. The pipeline provides both the ensemble value and the individual algorithm prediction affinities from each model. Dissimilarity analysis integrates Smith-Waterman alignments against the reference proteome, with a cutoff of dissimilarity metric >0.75 applied to identify high dissimilarity neoantigens, enhancing identification of immunogenic peptides. The workflow

is provided as an open-source tool, implemented in R, designed for Linux, and utilizes the "mclapply" function for parallelization.

NeoFuse [53] was proposed as the first fully unified workflow for the prediction of fusion neoantigens from tumor RNA-seq data. The unique characteristic of this tool lies in the integration of the preprocessing steps required for fusion transcript prediction in an end-to-end pipeline, which yields IC50 annotation for each neoantigen, percentile rank, confidence score, binding HLA type, expression of both HLA genes and fusion. Additionally, it identifies premature stop codons that could lead to nonsense-mediated decay of the fusion transcript. NeoFuse is a command line tool with five modules. Initially, data is imported in FASTQ format, and the first module performs HLA Class I typing using OptiType [54]. The second module utilizes Arriba [55] to predict fusion peptides, while the binding affinity of fusion peptides to HLA types is predicted by MHCflurry [23, 56] in the third module. Afterward, the fourth module quantifies gene expression levels as transcripts per million by utilizing both STAR [57] and featureCounts [58]. The fifth and final module filters and prioritizes based on the binding affinity and confidence score resulting from the fourth step. In this way, it produces a set of peptides that indicate potential fusion neoantigens. The tool is provided through two major container technologies, Docker and Singularity.

DeepHLApan [59] in contrast with the other methods is not a unified neoantigen prediction pipeline since it requires some preprocessing steps, yet it provides the distinctive feature of an end-to-end deep learning scheme for neoantigen prediction based on peptide-HLA binding and the immunogenicity of the complex. Specifically, DeepHLApan utilizes recurrent neural networks (RNN) and consists of two models, the first for predicting the probability of the peptide binding to the HLA in the tumor cell membrane and the second immunogenicity model for predicting the capacity of the peptide-HLA complex to induce T-cell activation. The immunogenicity score is used as a filter to rank the binding prediction score to yield a high confidence neoantigen identification. The model works only for HLA Class I neoantigens (A, B, and C alleles). Moreover, preprocessing steps are required since the model's input data should be in CSV format with the columns' heads being "Annotation, HLA, Peptide." DeepHLApan is provided as a ready-to-use model through a web platform or as a docker image.

pVACtools [60–62] is not a pipeline itself but a toolkit with several modules that can be integrated into one workflow to create an end-to-end neoantigen prediction tool. It is a modularized toolkit that provides the independent use of its module while facilitating multiple input types. It can also be integrated with external tools. The main feature of this toolkit is pVACseq, a pipeline for identifying and prioritizing neoantigens from a VCF

file, which can be coupled with the pVACviz GUI for the visualization and selection of data resulting from pVACseq. The pVACbind is used for FASTA files, while the pVACfuse is utilized for neoantigen prediction in gene fusions. Another tool is the pVACvector, which is employed to optimize the design of DNA-based cancer vaccines that prevent high-affinity junction neoantigens. Finally, the pVACapi offers a Rest-API for the pVACtools suite.

NextNEOpi [63] is a fully automated pipeline for neoantigen prediction with some special characteristics, such as quantification of neoepitope and patient-specific features associated with tumor immunogenicity and response to immunotherapy. NextNEOpi is a command line tool that utilizes raw DNA and RNA sequencing data, and a list of known patients' HLA types can also be imported. The first step of the pipeline after sequencing data pre-processing is HLA typing for both Class I using the OptiType and HLA-HD [64] for Class II. Then, variant calling is performed with several different independent algorithms, and variants called by more than one tool are marked as having high confidence. All variants are then annotated by the VEP tool, and the pVACseq tool is used to predict neoantigens from SNVs and INDELs, whereas NeoFuse is used to predict neoantigens from gene fusions. For peptide-HLA Class I binding prediction, NextNEOpi uses by default netMHCpan [65], MHCFlurry [23], and NetMHCIIpan [22]. For peptide-HLA II binding prediction mixMHC2pred [66] is employed. MiXCR [67] is used to predict T- and B-cell receptor repertoires, while clonality, tumor mutational burden, and CSiN scores are computed for the individual neoantigens and samples. NextNEOpi is implemented in NextFlow, providing reproducibility and scalability as a user-friendly tool.

NeoSplice [68] is another neoantigen prediction method, yet its unique characteristic is the use of splice variants. This method utilizes RNA-seq data as input and generates tumor-specific k-mers by comparing tumor cells with normal cells. Hence it identifies k-mer sequences abundant in the tumor transcriptome but rare in normal cells. Then, splice variant transcripts are predicted by constructing a splice graph using tumor cell RNA-seq data. Tumor-specific k-mers identified in the first step are then mapped to these splice variant transcripts. Annotations from Gencode are utilized to ascertain if the novel splice occurs within a protein-coding region and to determine the reading frame of the transcript. Finally, it translates novel splice junctions found within each splice variant transcript into peptide sequences based on the inferred open reading frame. Following translation, HLA Class I-binding affinity prediction is carried out on these peptide sequences employing NetMHCpan-4.0 [65] to identify regions that may produce neoepitopes. This tool is provided as a command line tool implemented in Python 2 while also shipping in a docker image.

Seq2Neo [69] provides a one-stop solution pipeline for neoantigen immunogenicity prediction and specifically for neoepitope feature prediction through raw sequencing data. The major distinctive characteristic of this pipeline is the use of a convolutional neural network (CNN) that predicts the immunogenicity of neoepitopes. Seq2Neo is a command line tool that automates workflows for predicting immunogenic peptides. It integrates mutation labeling, HLA typing, and HLA affinity-binding prediction tools, along with a (CNN)-based model for immunogenicity prediction. The workflow begins with importing raw sequencing data in FASTQ, SAM, or BAM format and then selecting the workflow of interest. For point mutation and INDEL detection, Mutect2 [70] was utilized, while for gene fusion detection STAR-Fusion [71] was employed. VCF format was used for the somatic variant data generated. HLA-HD is used for MHC genotyping, and ANNOVAR [72] or Agfusion [73] were utilized to annotate somatic variants to identify mutated peptides. Seq2Neo uses NetMHCpan for peptide-HLA-binding affinity prediction, while TPMCalculator [74] was used to detect gene expression and NetCTLpan [19] to obtain TAP transport efficiency. The tool outputs various peptide features, aiding in neoantigen prediction and immunogenicity assessment. Seq2Neo is provided as a Conda package or a docker image.

PGNneo [75] is another unique pipeline that performs neoantigen prediction in noncoding regions based on proteogenomics. The overall computational framework of PGNneo comprises the following components. First, there is noncoding somatic variant calling and HLA typing, this involves using paired tumor and normal samples for somatic variant calling, filtering out low-quality mutations, and extracting noncoding mutations. HLA typing is determined based on RNA-seq data from tumor samples. Second, nucleotide sequences are obtained and translated into proteins via six-frame translation. Tumor mutated peptides are extracted, and a customized protein database is constructed by combining these mutated protein sequences with reference proteins. Third, variant peptide identification involves filtering resulting peptides using MS datasets, providing evidence for their presence at protein levels and their binding to MHC molecules. Finally, neoantigen prediction and selection are conducted. Candidate neoantigens are predicted based on peptides and HLA types using NetMHCpan 4.1 [22]. These candidates undergo filtering using the dbPepNeo 2.0 database, which contains 746 experimental immunogenic peptides as a reference. The tool is provided both in a command line version and GUI versions, while for its implementation, Python, R, Java, and Perl were used.

NeoMUST [76] employs multitask learning, representing a novel approach to neoantigen prediction. The primary task of the model is neoantigen presentation classification, while the secondary task is binding affinity prediction between HLA Class I molecules

and eluted peptides. It effectively captures and utilizes task-specific information from both tasks, identifying similarities and distinctions to enhance performance. Additionally, it optimizes individual loss functions to balance the two tasks effectively while significantly reducing training time and enhancing scalability for large datasets. Although it is not an end-to-end pipeline, it features some novel capabilities. The model is available either as a Conda package or a docker image.

ImmuneMirror [77] is another recent method that provides an integrative pipeline for neoantigen prediction enhanced by machine learning. The machine learning model was constructed utilizing the balanced random forest algorithm to predict neoantigens. It integrates multiple biological features pertinent to neoantigen processes, including biogenesis, transportation, presentation, and T-cell recognition (such as agretopicity, foreignness, hydrophobicity, binding stability, peptide processing, and transportation scores). This machine learning model was then integrated into the ImmuneMirror bioinformatics pipeline, which also operates as a web server for predicting and prioritizing neoantigens from multiomics sequencing data. The pipeline accepts raw FASTQ reads as input, while the web server requires VCF files containing somatic mutations. The web server produces a visual report that incorporates the following: tumor mutational burden (TMB), HLA types, neoantigen load for HLA Class I and II, mismatch repair (MMR) status, germline and somatic mutations, ImmuneMirror prediction score, and IPRES gene expression signature.

GraphMHC [78] is one of the most recent approaches for neoantigen prediction, utilizing a graph neural network applied to molecular structure to simulate the binding between peptide and MHC proteins. The pipeline begins by converting HLA into MHC amino acid sequences. Next, both MHC and peptide sequences are transformed into SMILES structures using the RDKit 2022.03.2 library. Then, these two SMILES strings are combined using non-bond notation. Afterward, the combined structure is converted into a molecular structure using RDKit, ensuring that any omitted hydrogen atoms are included. Following this, the molecular structure is transformed into a graph structure using the RDKit library, allowing for the encoding of vectors and matrices. Each feature is encoded using one-hot encoding and assembled into a sparse matrix. Finally, the graph dataset is converted using the PyTorch Geometric (PyG) 2.1.0 library.

Table 1 summarizes all the methods discussed. The table provides the name of each tool, the intended function of the software and specifies the input data utilized in neoantigen prediction. Additionally, it provides neoantigen classification, the method employed to evaluate the binding affinity between the neoantigen and HLA molecules and the outcomes obtained from the analysis. A link to

Table 1
Computational tools, pipelines, and workflows for neoantigen prediction

Name	Purpose	Input type	HLA Class	HLA binding affinity method	Output	Source	Year of publication	Last Update
CloudNeo	Cloud-based pipeline to streamline neoantigen prediction	VCF, BAM	HLA Class I	NetMHCpan 3.0	Peptide subsequences and MHC-binding affinity scores for all six HLA types	https://github.com/TheJacksonLaboratory/CloudNeo/tree/master	June 2017	2019
Antigen.garnish	An R package to predict neoantigen immunogenicity based on dissimilarity to self-proteome	VCFs or ensemble transcript IDs or HGVS-style cDNA annotations	HLA Class I & II	Ensemble affinity score utilizing: NetMHCI/II, netMHCI/IIpan, MHCflurry, and MHCnuggets	Ensemble binding affinity and peptide immunogenicity	https://github.com/andrewrech/antigen.garnish	October 2019	2022
NeoFuse	An automated workflow for the identification of fusion neoantigens	FASTQ files	HLA Class I	MHCflurry	Neoantigen prioritization based on IC50 binding affinity, confidence score, and annotation of each neoantigen	https://github.com/icbi-lab/NeoFuse	November 2019	2022
DeepHLApan	A deep learning method for predicting neoantigens based on peptide-HLA binding and the immunogenicity of the complex	CSV file with annotations, HLA alleles and peptides	HLA Class I & II	Recurrent Neural Network-based approach integrating HLA-peptide binding and pHLA immunogenicity	Neoantigen prediction and immunogenicity	https://github.com/jiujiezz/deephlapan	November 2019	2022
pVACtools	A modularized toolkit for neoantigen predictions with an interactive display for review by the end user. Highly compatible with external tools	BAM, VCF, FASTA	HLA Class I & II	Utilizes the algorithms supported in pVACseq	Neoantigen prediction and ranking	https://github.com/griffithlab/pVACtools	March 2020	2024

NextNEOpi	Automated bioinformatics pipeline for predicting tumor neoantigens from raw DNA and RNA sequencing	Raw FASTQ or BAM files	HLA Class I & II	NetMHCpan 4.0 and MHCflurry 2.0	Tumor mutational burden, canonical neontigens, fusion neoantigens	https://github.com/icbi-lab/nextNEOpi	November 2021	2023
NeoSplice	An end-to-end workflow for splice variant neoantigen prediction	BAM files	HLA Class I & II	NetMHCpan 4.0	HLA binding prediction, with predicted binders representing putative splice variant neoantigens	https://github.com/pirl-unc/NeoSplice	May 2022	2024
Seq2Neo	A comprehensive pipeline to predict the immunogenicity of neoepitopes derived from somatic DNA alterations, aiding in cancer immunotherapy	Files in FASTQ, SAM and BAM format	HLA Class I	NetMHCpan 4.1 & NetMHCIIpan-4.0	Binding affinity, TAP transport efficiency, gene expression, and immunogenicity score	https://github.com/XSLiuLab/Seq2Neo	October 2022	2023
PGNneo	An integrated pipeline for neoantigen prediction in noncoding region based on proteogenomics	FASTQ and raw MS data	HLA Class I	NetMHCpan 4.1	Neoantigen prediction and selection	https://github.com/tanxiaoxiu/PGNneo	March 2023	2023
NeoMUST	A deep learning model utilizing multitask learning	CSV file with columns "hla,peptide," BLOSUM62 file and a CSV file for HC_pseudo-sequences with columns "allele, sequence"	HLA Class I	Multitask learning (MTL) approach to predict the neoantigen presentation as its main task and the neoantigen–MHC binding as an auxiliary task	Binding affinity, eluted peptide, and the rank of the peptide	https://github.com/Freshwind-Bioinformatics/NeoMUST	January 2024	2024
ImmuneMirror	A web server for an integrative pipeline for neoantigen prediction leveraging machine learning	FASTQ for the CLI and VCF for the web app	HLA Class I & II	pVACtools	TMB, HLA types, neoantigen load for HLA Class I and II, MMR status, germline and somatic mutations,	https://github.com/weidai2/ImmuneMirror/	February 2024	2024

(continued)

Table 1
(continued)

Name	Purpose	Input type	HLA Class	HLA binding affinity method	Output	Source	Year of publication	Last Update
					ImmuneMirror prediction score, and IPRES gene expression signature			
GraphMHC	A Graph Neural Network utilizing molecular structure to model the binding affinity for neoantigen prediction-based	CSV HLA type, peptide sequence	HLA Class I and II	Graph neural network for MHC-peptide-binding prediction	Neoantigen binding	https://github.com/recognizability/GraphMHC	March 2024	2024

VCF Variant call format, *BAM* binary alignment map, *HGVS* human genome variation society, *CSV* comma-separated values, *SAM* sequence alignment/map format, *HLA* human leukocyte antigens, *MHC* major histocompatibility complex, *MS* mass spectrometry, *CLI* command-line interface, *TBM* tumor mutation burden, *MMR* DNA mismatch repair, *IPRES* A transcriptional signature related to innate anti-PD-1 resistance, *TAP* transporter associated with antigen processing, *IC50* half-maximal inhibitory concentration

the source code or user interface of the software is also provided along with its publication date and the date of its last update.

The above methods provide only an overview of the distinctive utilities offered by available tools and pipelines. However, there are several other methods available for neoantigen prediction. The choice of method depends on factors such as the specific use, the user's level of bioinformatics expertise, and the ease of pipeline implementation. Therefore, users should consider both the input and output data of each pipeline based on their needs. Additionally, users can choose among command-line interface (CLI) tools or web applications with user-friendly interfaces, depending on their proficiency in utilizing informatics tools. It is important to note that a direct comparison of the prediction accuracy of tools can only be made when the prediction endpoint of the pipeline is the same.

In the last decade, deep learning models have flourished due to their high prediction accuracy across several fields. Thus, they have been widely adopted in biomedical research. This trend is particularly evident in the field of neoantigen prediction, where deep learning methods have been introduced for binding affinity prediction. While machine learning models were predominantly utilized for neoantigen-peptide binding, there has been a noticeable transition in many pipelines toward deep learning methods. This shift is strongly correlated with the continuous expansion of available training data and the emergence of additional features. As a result, the complexity of the data is increasing, favouring deep learning models due to their enhanced capacity to capture and process this wealth of information compared to traditional machine learning models.

5 Neoantigen Prediction in Mouse Models

In silico prediction of neoantigens represents a pivotal phase in unlocking the therapeutic potential of cancer immunotherapy. As described in the previous section, a plethora of software for neoantigen discovery is available. However, these pipelines are mainly tailored to human data with a focus on predicting the binding affinity between epitope and HLA. Nevertheless, models specific to murine systems are crucial for facilitating in vivo experimentation and further translation of immunotherapies into clinical practice. The availability of these pan-specific software solutions remains limited, posing a significant challenge in preclinical immunotherapy research. Consequently, some human-centric software platforms have undergone adaptation to include binding affinity predictions for mouse MHC. Moreover, efforts have also been made to develop murine-specific models aimed at bridging this gap in experimental settings.

When evaluating the collection of available software tools, it is important to recognize that neoantigen presentation and recognition by T cells entail a complex process comprising various steps. Many existing neoantigen prediction tools primarily focus on predicting the binding affinity between the epitope and the MHC molecule while overlooking other critical steps, leading to a high false positive rate of predicted epitopes [64]. Emerging software solutions are now considering these additional steps to yield more robust predictions. First, the different software platforms specializing in binding affinity prediction within murine models will be examined. These tools exhibit variations in training data modalities, training methodologies, and input data types. While some tools exclusively predict binding affinity for user-identified neoepitopes, others offer end-to-end platforms capable of processing RNA-seq data to predict neoantigens directly. These latter tools enable users to input raw data directly without having to create and apply variant calling pipelines. In terms of training data, it is common to utilize either binding affinity data or mass spectrometry-eluted ligands. However, studies have demonstrated that combining both input data types enhances predictive performance [22]. Additionally, predictive models have transitioned from earlier methodologies, such as support vector machine regression [79] or profiles [80], to more advanced approaches like ANNs and RNNs, which have shown superior performance.

Among the reviewed software solutions, only two are explicitly designed for murine models. The first, NetH2Pan [81], employs an ANN architecture to predict binding affinity. It performs the prediction based on user-provided peptides, leveraging both binding affinity and eluted ligand data during training. Conversely, NAP-CNB [9] operates as an end-to-end platform, using RNA-seq for neoantigen prediction. The tool integrates a variant calling pipeline that returns SNVs and INDELs unique to the tumor. This method implements a more advanced neural network with long-short-term memory (LSTM) units, albeit trained solely on binding affinity data. Additionally, several software platforms initially developed for human data have been adapted to incorporate murine H2 alleles. Examples of such software include NetMHC [25], NetMHCpan [65], and MHCflurry [23]. Both NetMHC and NetMHCpan utilize an ANN architecture and were trained on binding affinity and eluted ligand data. Moreover, they accept user-generated tumor-specific peptides as input. However, NetMHC employs an allele-specific training approach, while NetMHCpan adopts a "pan-specific" strategy, combining information from both data modalities and diverse MHC molecules into a unified network. The authors reported that the novel training strategy employed by NetMHCpan enhances predictive accuracy. In contrast, MHCflurry uses binding affinity and eluted ligand data in a more sequential manner. The method initially conducts

binding predictions using an ensemble of ANNs trained on binding affinity data and subsequently integrates mass spectrometry data into another ensemble of ANNs to account for the antigen processing steps, particularly focusing on proteasomal cleavage. The outputs of these two models are then aggregated to generate a comprehensive presentation score. Furthermore, certain tools seek to bridge the gap between RNA-seq data and tumor-specific peptides by integrating some of the prediction methods mentioned above with variant calling pipelines, offering end-to-end solutions for neoantigen prediction. Examples include Epi-Seq [82], a bioinformatics pipeline utilizing NetMHC, and pVAC-Seq [62], which incorporates various prediction methods like NetMHC or MHCflurry. A general summary of these methods can be found in Table 2.

Newly developed methods are taking into consideration additional steps of the neoantigen processing pipeline, aiming to reduce the occurrence of false positives. One such tool, DeepNeo [83], a neural network-based tool, integrates predictions on MHC binding affinity and T-cell reactivity, a crucial factor for the success of neoantigen vaccines. The tool accepts peptide sequences from both mouse and human data and generates a binary prediction for MHC binding alongside a quantification of T-cell reactivity. While the prediction of T-cell reactivity holds promise for designing more effective neoantigen-based treatments, it is worth noting that the authors do not provide validation of the tool's performance on murine data in the paper. Another recently introduced tool, Neo-Intiline [84], similarly accounts for various stages of peptide presentation and recognition. The tool is designed to be used with WGS data. The tool's optimal performance is observed when analyzing melanoma data, although its applicability extends to any relevant dataset of interest.

Although these methods have demonstrated efficacy in silico, in vivo validations are imperative to assess their real-world performance. Both NetH2pan and NAP-CNB, validated in clinical settings, have proven effective in neoantigen discovery [85].

6 Technical Challenges and Future Improvements

The field of neoantigen prediction has evolved significantly, propelled by advancements in computational biology, high-throughput sequencing technologies, and the integration of machine learning approaches. Despite these advancements, several technical challenges persist, and addressing these challenges is crucial for enhancing the predictive accuracy and clinical utility of neoantigen prediction methods.

Table 2
Methods for MHC Class I-binding affinity prediction

Name	Input type	Encoding	Type of data for training	Prediction method	Mouse-specific?	Source	Year of publication
NetH2Pan	Peptides or protein sequence	BLOSUM encoding	BA + EL	ANN	Yes	Web server: https://services.healthtech.dtu.dk/services/NetH2pan-1.0/	March, 2018
NAP-CNB	RNA-seq or peptides	One-hot encoding	BA	LSTM units and dense units	Yes	Web server: https://biocomp.cnb.csic.es/NeoantigensApp/	May, 2021
NetMHC 4.0	Peptides or protein sequence	BLOSUM encoding	BA + EL	ANN	No	Web server: https://services.healthtech.dtu.dk/services/NetMHC-4.0/	February, 2016
NetMHCpan 4.1	Peptides or protein sequence	BLOSUM encoding	BA + EL	ANN	No	Web server: https://services.healthtech.dtu.dk/services/NetMHCpan-4.1/	July, 2020
MHCflurry 2.0	Peptides or proteins	45-mer representation and BLOSUM	BA + EL	ANN	No	Command line tool: https://github.com/openvax/mhcflurry	July, 2020
Epi-seq	RNA-seq	–	–	NetMHC	No	Software available at https://dna.engr.uconn.edu/?page_id=470	October, 2014
pVAC-Seq	VCF file	–	–	Different softwares available	No	Software available at https://github.com/griffithlab/pVACtools	January, 2016

VCF variant call format, *BLOSUM* BLOcks SUbstitution Matrix, *BA* binding affinity, *EL* eluted ligand, *ANN* artificial neural network, *LSTM* long short-term memory

6.1 Technical Challenges

The main challenges we have identified are:

1. High false positive rates: One of the enduring challenges in neoantigen prediction is the high rate of false positives. Many predicted neoantigens are not genuinely immunogenic, which can lead to inefficient or ineffective therapeutic strategies. This challenge stems primarily from the limitations in accurately modeling the complex interplay of factors that contribute to the immunogenicity of neoantigens, such as peptide-MHC-binding affinity, TCR recognition, and the expression and presentation dynamics in tumor microenvironment.
2. HLA allelic diversity: The genetic diversity of HLA alleles poses a significant challenge due to its impact on binding affinity predictions. Current prediction tools often have reduced accuracy for less common HLA alleles, which are underrepresented in training datasets. This limitation affects the generalizability of prediction models across different populations. In addition, most tools are specific to MHC Class I molecules, although it has been demonstrated that MHC Class II is also essential for effective antitumor immune responses.
3. Integration of epitope processing: Neoantigen prediction tools primarily focus on the binding affinity of peptides to MHC molecules. However, the entire process of antigen presentation, including proteasomal processing, transport by TAP proteins, and trimming by ER aminopeptidases, significantly influences the presence of peptides on the cell surface. The lack of comprehensive integration of these steps can lead to inaccuracies in predicting true neoantigens.
4. Scalability and computational efficiency: As genomic datasets grow in size and complexity, the computational demands of neoantigen prediction also increase. Scalability and efficiency become critical, especially for real-time or near-real-time analysis in clinical settings. Many existing tools require substantial computational resources, which can be a barrier to routine clinical use.

6.2 Future Improvements

As future improvements to be accomplished, we propose the following:

1. Enhanced machine learning models: Future advancements should include the development of more sophisticated machine learning models integrating multiple aspects of antigen presentation and immune recognition. Deep learning approaches that can learn complex patterns from large datasets may offer improvements in predicting the immunogenicity of neoantigens beyond mere peptide-MHC binding and development of neoantigen-based therapies.

2. Incorporation of tumor microenvironment factors: Incorporating data from the tumor microenvironment, such as cytokine profiles, immune infiltration, and checkpoint expression, could enhance the prediction of neoantigen immunogenicity. Understanding the interaction between neoantigens and the tumor microenvironment will aid in prioritizing neoantigens that are more likely to elicit a robust immune response.
3. Expanding training datasets: To address the issue of HLA diversity, it is essential to expand training datasets to include a broader array of HLA types, particularly those that are less common globally. This expansion would improve the accuracy of the model and its applicability to diverse populations.
4. Integrative multiomics approaches: Future tools should aim to integrate multi-omics data, including genomics, transcriptomics, and proteomics, to provide a more holistic view of neoantigen presentation and potential immunogenicity. This integration will help in understanding the complex dynamics of cancer biology and immune responses.
5. Cloud-based platforms and real-time analysis: Developing cloud-based platforms that can perform real-time analysis of neoantigen predictions would significantly benefit clinical applications. Such platforms should be designed to handle large-scale data efficiently, providing accessible and rapid insights for personalized cancer immunotherapy.

7 Discussion and Conclusions

The prediction of neoantigens represents a cornerstone in the development of personalized cancer immunotherapies. It leverages the power of computational biology, genomics, and immunology to identify tumor-specific antigens that can be targeted by the immune system, offering a highly personalized approach to cancer treatment. The insights gained from this research area are critical in guiding the design of vaccines and cell-based therapies that have the potential to significantly improve patient outcomes.

Throughout this chapter, various computational methods and tools developed for neoantigen prediction have been explored. These tools have evolved from basic sequence alignment techniques to sophisticated machine learning models that predict peptide-MHC-binding affinities and assess immunogenic potential. The integration of deep learning has particularly enhanced the accuracy and predictive power of these tools, reflecting broader trends in biomedical research where advanced computational methods are increasingly pivotal.

However, despite these technological advancements, several challenges remain. The prediction of neoantigens still contends

with issues such as high false positive rates, limited understanding of the immunogenicity landscape, and the need for better integration of comprehensive antigen processing pathways. Moreover, the diversity of HLA alleles presents a significant hurdle in achieving universally applicable prediction tools, necessitating ongoing efforts to expand and diversify the training datasets used in model development.

Looking forward, the field of neoantigen prediction is poised for transformative growth. Key areas for future improvement include the development of integrative multi-omics platforms that can provide a more complete picture of tumor immunogenicity and the microenvironmental factors influencing immune recognition. Additionally, the expansion of machine learning models to include more diverse data types and training sets will enhance the accuracy and applicability of predictions across different populations and cancer types.

Ultimately, the integration of neoantigen prediction into clinical practice promises to revolutionize cancer immunotherapy. By tailoring treatments to the specific immunogenic landscape of the tumor of each patient, neoantigen prediction paves the way for more effective and less toxic therapies. It holds the promise of turning the immune system into a precise tool for targeting cancer, fundamentally changing the way we approach cancer treatment and heralding a new era of precision oncology. As we continue to refine and improve computational methods for neoantigen prediction, we move closer to realizing the full potential of immunotherapy in providing durable and potent cancer treatments.

Acknowledgments

We would like to acknowledge, in memory of Sotiris, his early contributions to this work. His presence and spirit remain fondly remembered. This work has been supported by the Aragon Government (Group B29_23R), the Instituto de Salud Carlos III (ISCIII) grant PI22/01938, and by ASPANOA Foundation. This work is part of the CNS2022-136176 action, financed by MCIN/AEI/10.13039/501100011033 and for the European Union «Next Generation EU»/PRTR. This publication is based upon work from COST Action IMMUNO-model, CA21135, supported by COST (European Cooperation in Science and Technology).

References

1. Jhunjhunwala S, Hammer C, Delamarre L (2021) Antigen presentation in cancer: insights into tumour immunogenicity and immune evasion. Nat Rev Cancer 21:298–312. https://doi.org/10.1038/s41568-021-00339-z

2. Smith CC, Selitsky SR, Chai S et al (2019) Alternative tumour-specific antigens. Nat Rev Cancer 19:465–478. https://doi.org/10.1038/s41568-019-0162-4

3. Turajlic S, Litchfield K, Xu H et al (2017) Insertion-and-deletion-derived tumour-specific neoantigens and the immunogenic phenotype: a pan-cancer analysis. Lancet Oncol 18:1009–1021. https://doi.org/10.1016/S1470-2045(17)30516-8
4. Li J, Xiao Z, Wang D et al (2023) The screening, identification, design and clinical application of tumor-specific neoantigens for TCR-T cells. Mol Cancer 22:141. https://doi.org/10.1186/s12943-023-01844-5
5. Neefjes J, Jongsma MLM, Paul P, Bakke O (2011) Towards a systems understanding of MHC class I and MHC class II antigen presentation. Nat Rev Immunol 11:823–836. https://doi.org/10.1038/nri3084
6. Marty Pyke R, Thompson WK, Salem RM et al (2018) Evolutionary pressure against MHC class II binding cancer mutations. Cell 175:416–428.e13. https://doi.org/10.1016/j.cell.2018.08.048
7. Xie N, Shen G, Gao W et al (2023) Neoantigens: promising targets for cancer therapy. Signal Transduct Target Ther 8:9. https://doi.org/10.1038/s41392-022-01270-x
8. Méndez-Pérez A, Acosta-Moreno AM, Wert-Carvajal C et al (2023) Unraveling the power of NAP-CNB'S machine learning-enhanced tumor neoantigen prediction. Life 13:RP95010. https://doi.org/10.7554/eLife.95010.1
9. Wert-Carvajal C, Sánchez-García R, Macías JR et al (2021) Predicting MHC I restricted T cell epitopes in mice with NAP-CNB, a novel online tool. Sci Rep 11:10780. https://doi.org/10.1038/s41598-021-89927-5
10. Borden ES, Buetow KH, Wilson MA, Hastings KT (2022) Cancer neoantigens: challenges and future directions for prediction, prioritization, and validation. Front Oncol 12:836821. https://doi.org/10.3389/fonc.2022.836821
11. Griffith M, Miller CA, Griffith OL et al (2015) Optimizing cancer genome sequencing and analysis. Cell Syst 1:210–223. https://doi.org/10.1016/j.cels.2015.08.015
12. Bhagwate AV, Liu Y, Winham SJ et al (2019) Bioinformatics and DNA-extraction strategies to reliably detect genetic variants from FFPE breast tissue samples. BMC Genomics 20:689. https://doi.org/10.1186/s12864-019-6056-8
13. Holtsträter C, Schrörs B, Bukur T, Löwer M (2020) Bioinformatics for cancer immunotherapy. Methods Mol Biol 2120:1–9. https://doi.org/10.1007/978-1-0716-0327-7_1
14. Koboldt DC (2020) Best practices for variant calling in clinical sequencing. Genome Med 12:91. https://doi.org/10.1186/s13073-020-00791-w
15. Gfeller D, Guillaume P, Michaux J et al (2018) The length distribution and multiple specificity of naturally presented HLA-I ligands. J Immunol 201:3705–3716. https://doi.org/10.4049/jimmunol.1800914
16. Osterbye T, Nielsen M, Dudek NL et al (2020) HLA class II specificity assessed by high-density peptide microarray interactions. J Immunol 205:290–299. https://doi.org/10.4049/jimmunol.2000224
17. Saha I, Mazzocco G, Plewczynski D (2013) Consensus classification of human leukocyte antigen class II proteins. Immunogenetics 65:97–105. https://doi.org/10.1007/s00251-012-0665-6
18. Saxová P, Buus S, Brunak S, Keşmir C (2003) Predicting proteasomal cleavage sites: a comparison of available methods. Int Immunol 15:781–787. https://doi.org/10.1093/intimm/dxg084
19. Stranzl T, Larsen MV, Lundegaard C, Nielsen M (2010) NetCTLpan: pan-specific MHC class I pathway epitope predictions. Immunogenetics 62:357–368. https://doi.org/10.1007/s00251-010-0441-4
20. Baxter-Lowe LA (2021) The changing landscape of HLA typing: understanding how and when HLA typing data can be used with confidence from bench to bedside. Hum Immunol 82:466–477. https://doi.org/10.1016/j.humimm.2021.04.011
21. Barker DJ, Maccari G, Georgiou X et al (2023) The IPD-IMGT/HLA database. Nucleic Acids Res 51:D1053–D1060. https://doi.org/10.1093/nar/gkac1011
22. Reynisson B, Alvarez B, Paul S et al (2020) NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. Nucleic Acids Res 48:W449–W454. https://doi.org/10.1093/nar/gkaa379
23. O'Donnell TJ, Rubinsteyn A, Laserson U (2020) MHCflurry 2.0: improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. Cell Syst 11:42–48.e7. https://doi.org/10.1016/j.cels.2020.06.010
24. Vita R, Mahajan S, Overton JA et al (2019) The immune epitope database (IEDB): 2018 update. Nucleic Acids Res 47:D339–D343. https://doi.org/10.1093/nar/gky1006
25. Lundegaard C, Lamberth K, Harndahl M et al (2008) NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey

MHC class I affinities for peptides of length 8-11. Nucleic Acids Res 36:W509–W512. https://doi.org/10.1093/nar/gkn202

26. Hellman LM, Yin L, Wang Y et al (2016) Differential scanning fluorimetry based assessments of the thermal and kinetic stability of peptide-MHC complexes. J Immunol Methods 432:95–101. https://doi.org/10.1016/j.jim.2016.02.016
27. Blaha DT, Anderson SD, Yoakum DM et al (2019) High-throughput stability screening of neoantigen/HLA complexes improves immunogenicity predictions. Cancer Immunol Res 7:50–61. https://doi.org/10.1158/2326-6066.CIR-18-0395
28. Zhang H, Lund O, Nielsen M (2009) The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: application to MHC-peptide binding. Bioinforma 25:1293–1299. https://doi.org/10.1093/bioinformatics/btp137
29. Paul S, Grifoni A, Peters B, Sette A (2019) Major histocompatibility complex binding, eluted ligands, and immunogenicity: benchmark testing and predictions. Front Immunol 10:3151. https://doi.org/10.3389/fimmu.2019.03151
30. Gfeller D, Bassani-Sternberg M (2018) Predicting antigen presentation-what could we learn from a million peptides? Front Immunol 9:1716. https://doi.org/10.3389/fimmu.2018.01716
31. Marrer-Berger E, Nicastri A, Augustin A et al (2024) The physiological interactome of TCR-like antibody therapeutics in human tissues. Nat Commun 15:3271. https://doi.org/10.1038/s41467-024-47062-5
32. Zvyagin IV, Tsvetkov VO, Chudakov DM, Shugay M (2020) An overview of immunoinformatics approaches and databases linking T cell receptor repertoires to their antigen specificity. Immunogenetics 72:77–84. https://doi.org/10.1007/s00251-019-01139-4
33. Ji H, Wang X-X, Zhang Q et al (2024) Predicting TCR sequences for unseen antigen epitopes using structural and sequence features. Brief Bioinform 25:bbae210. https://doi.org/10.1093/bib/bbae210
34. Roudko V, Greenbaum B, Bhardwaj N (2020) Computational prediction and validation of tumor-associated neoantigens. Front Immunol 11:27. https://doi.org/10.3389/fimmu.2020.00027
35. Shah RK, Cygan E, Kozlik T et al (2023) Utilizing immunogenomic approaches to prioritize targetable neoantigens for personalized cancer immunotherapy. Front Immunol 14:1301100. https://doi.org/10.3389/fimmu.2023.1301100
36. Martin MV, Aguilar-Rosas S, Franke K et al (2024) The neo-open reading frame peptides that comprise the tumor framome are a rich source of neoantigens for cancer immunotherapy. Cancer Immunol Res 12:759. https://doi.org/10.1158/2326-6066.CIR-23-0158
37. Rasmussen M, Fenoy E, Harndahl M et al (2016) Pan-specific prediction of peptide-MHC class I complex stability, a correlate of T cell immunogenicity. J Immunol 197:1517–1524. https://doi.org/10.4049/jimmunol.1600582
38. Nielsen M, Lundegaard C, Lund O, Keşmir C (2005) The role of the proteasome in generating cytotoxic T-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. Immunogenetics 57:33–41. https://doi.org/10.1007/s00251-005-0781-7
39. Peters B, Bulik S, Tampe R et al (2003) Identifying MHC class I epitopes by predicting the TAP transport efficiency of epitope precursors. J Immunol 171:1741–1749. https://doi.org/10.4049/jimmunol.171.4.1741
40. Zhao W, Sher X (2018) Systematically benchmarking peptide-MHC binding predictors: from synthetic to naturally processed epitopes. PLoS Comput Biol 14:e1006457. https://doi.org/10.1371/journal.pcbi.1006457
41. García-Mulero S, Fornelino R, Punta M et al (2023) Driver mutations in GNAQ and GNA11 genes as potential targets for precision immunotherapy in uveal melanoma patients. Onco Targets Ther 12:2261278. https://doi.org/10.1080/2162402X.2023.2261278
42. Ragone C, Cavalluzzo B, Mauriello A et al (2024) Lack of shared neoantigens in prevalent mutations in cancer. J Transl Med 22:344. https://doi.org/10.1186/s12967-024-05110-0
43. Hsiue EH-C, Wright KM, Douglass J et al (2021) Targeting a neoantigen derived from a common *TP53* mutation. Science 371:eabc8697. https://doi.org/10.1126/science.abc8697
44. Hoyos D, Zappasodi R, Schulze I et al (2022) Fundamental immune–oncogenicity trade-offs define driver mutation fitness. Nature 606:172–179. https://doi.org/10.1038/s41586-022-04696-z
45. Choi J, Goulding SP, Conn BP et al (2021) Systematic discovery and validation of T cell targets directed against oncogenic KRAS mutations. Cell Rep Methods 1:100084. https://doi.org/10.1016/j.crmeth.2021.100084
46. Hartmaier RJ, Charo J, Fabrizio D et al (2017) Genomic analysis of 63,220 tumors reveals insights into tumor uniqueness and targeted cancer immunotherapy strategies. Genome