NetPolyEV: An epitope vaccine polytope optimizer

Chen Chen
Danmarks Tekniske Universitet

Abstract

Highly mutating viruses such as hepatitis C virus (HCV), coronavirus disease 2019 (COVID-19), influenza viruses (Flu), and respiratory syncytial virus (RSV) are calling for epitope-based vaccines (EV) and the construction of polytopes. Here, epitopes with broad HLA and pathogen diversity coverage can be included in a string-of-beads polypeptides vaccine construct. A challenge of such polytope constructs is the presence of neo-fusion epitopes.

Here, we propose PolyVaccine, to resolve this challenge. PolyVaccine CD8 epitopes (predicted or measured) as input and reports a polytope solution with minimal neo-epitope occurrence. The solution is obtained mainly implementing a Monte Carlo simulated annealing approach where both the order of the included CD8+ T cell epitopes and the spacers with flexible length and sequence are optimized, to minimize the number of predicted neo-fusion epitopes. The use of PolyVaccine was showcased on polytope vaccines for HCV, COVID-19, Flu, Smallpox and RSV, and was in all cases demonstrated to result in our constructs with zero neo-epitopes. PolyVaccine-1.0 is as a webserver at `https://services.healthtech.dtu.dk/services/NetPolyEV-1.0/`.

1 Introduction

Traditional vaccines contain potentially infectious material or virulence and have challenges to protect a broad population when the virus is in rapidly mutating. The T cell epitope-based vaccines that the epitopes are presented on the protein of the major histocompatibility complex(MHC) and thus can induce a potential T-cell mediated immune response offer an alternative way, and the epitopes will be chosen to target different antigens and alleles to maximize the immunogenicity[1–3]. The immune response is not efficient to be elicited when delivering the mixture of the separated epitopes, so concatenating the selected epitopes into a polytope vaccine which can potentially elicit a strong T cell reaction and increase the wanted immunogenicity is in high demand[4, 5].
Two main stages are needed for constructing a polytope vaccine: epitope selection and epitope assembly. First, the epitopes are selected to have broad alleles coverage and to target preferred antigens to maximize the immunogenicity. Due to the high polymorphism of HLA molecules, different persons will react to pathogen infection in a different way. Hence, T cell epitopes are needed to be selected to match the HLA alleles in the target population or person. Second, the selected epitopes from the first stage are assembled into polytope vaccine to minimize the neo-immunogenicity by optimizing the order of the epitopes and the linkers between pairwise epitopes[1, 3].
The main challenges in designing optimized polytope vaccine are determining optimal spacers or linkers with felxible length and the optimal order of the selected epitopes. For instance, there are 10! distinct polytope vaccines without considering spacers or linkers with 10 selected epitopes. Considering spacers with flexible length and sequence will increase the possibilities to trillions. The computational approaches hence have been proposed to address epitope assembly problem. For instance, traveling salesperson approach was implemented to find the optimal order of selected epitopes, and the spacer with flexible length was designed by

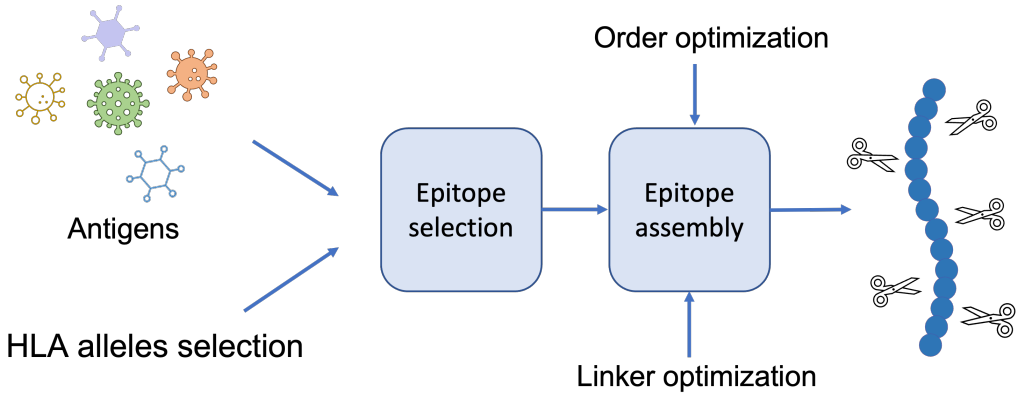

Figure 1: *Pipeline of designing polytope vaccine.*
The epitopes are selected to match a set of HLA alleles in a target population or person and assembled to the polytope vaccine to have less neo-immunogenicty by optimizing the order of the epitopes and the linkers at one procedure. The obained polytope vaccine will then be cleavaged in the antigen-presenting cells to recover the selected epitopes which will be presented on the MHC molecules.

applying lexicographical ordered optimization formulation[1, 2].
The prediction of presented epitopes of the CD8+ T cell epitopes-based vaccine in our work relies on NetMHCpan-4.1 which exploits tailored machine learning strategies to predict peptide-MHC class I binding affinity. Because the most selective step in the antigen presentation pathway is the binding between MHC molecules and antigenic peptides, the predicted peptides that bind to HLA class I alleles can potentially elicit the cytotoxic T-cell immune response. The input data of NetMHCpan-4.1 is a polytope vaccine in the fasta format, and the MHC-I alleles should be chosen. If the predicted peptide is strong or weak MHC binder will be informed based on %Rank score[6, 7]. The focus of this work is mainly on applying the monte carlo simulated annealing approach for cytotoxic T cell epitope assembly to make 0 neo-epitope achievable for the optimized polytope vaccines. Additionally, the big step greedy search is introduced to help reduce time consumption when constructing the vaccines. The order of selected epitopes and optimal spacers with flexible length and sequence can be optimized during the procedure. The polytope vaccines with 0 neo-epitope are constructed successfully for the viruses HCV, COVID-19, Flu, Smallpox and RSV by implementing our program. Our results indicate the 0 neo-epitope of the polytope vaccines is achievable.

2 Materials and methods

2.1 Epitopes and alleles data

We focused on constructing polytope vaccines that can potentially elicit CD8+ T cell immune response for hepatitis C virus (HCV), coronavirus disease 2019 (COVID-19), influenza viruses (Flu), Smallpox and respiratory syncytial virus (RSV). In Table 1, the selected HLA alleles and epitopes are shown. 10 most popular HLA class I alleles in European population are selected from the *Allele Frequency Net Database*[8] and the epitopes from HCV, COVID-19, Flu, Smallpox and RSV are choosen for matching those 10 alleles from *IEDB*[9].

Table 1: *The selected HLA class I alleles and epitopes.*

Alleles	HCV	COVID-19	Flu	Smallpox	RSV
HLA-A02:01	CINGVCWTV	YLQPRTFLL	GILGFVFTL	ALWDSKFFT	NPKASLLSL
HLA-A01:01	ALYDVVTKL	QYIKWPWYI	AIMDKNIIL	ILHDNVVTL	YLEKESIYY
HLA-A03:01	DLMGYIPLV	NYNYLYRLF	CTELKLSDY	KLLDKTSLV	AELDRTEEY
HLA-A24:02	GLQDCTMLV	LTDEMIAQY	LPFEKSTVM	LLYAFMYYL	ILNNPKASL
HLA-A11:01	CVNGVCWTV	FIAGLIAIV	ILRGSVAHK	RLYDYFTRV	IPYSGLLLV
HLA-B07:02	HSKKKCDEL	RLQSLQTYV	FMYSDFHFI	VLQDQLTMV	KMLKEMGEV
HLA-B51:01	ATDALMTGY	VLNDILSRL	LPFDKPTIM	YLAKLTALV	LAKAVIHTI
HLA-B08:01	GPRLGVRAT	SIIAYTMSL	NMLSTVLGV	ILKSLGFKV	QLLSSSKYT
HLA-B35:01	ARMILMTHF	KCYGVSPTK	VSDGGPNLY	NPVTIINEY	QVMLRWGVL
HLA-B44:02	ILAGYGAGV	KIADYNYKL	ELRSRYWAI	NLLCHIYSL	RARRELPRF

2.2 The prediction for peptide-MHC binding

NetMHCpan-4.1 was applied to predict binding of 9mer peptides (the most common length of HLA class I antigen presented peptides) contained within the polytope sequence. Default threshold %Rank < 0.5% for strong (SB) and %Rank < 2% and weak (WB) binders was applied[6, 7].

2.3 The big step of greedy search

Algorithm 1 comprises the BM function, which is responsible for producing a polytope vaccine. The acronym BM denotes epitope shuffling along with the insertion of a randomly selected space or linker, derived from the preferred linker input, between pairwise epitopes. The inserted space or linker consists of 0 to 5 amino acids selected randomly from the 20 standard amino acids. The BM function is introduced for generating both the initial polytope (I) and new polytope sequences at each iteration in the for loop. The cost function is the number of neo-epitopes of the polytope sequence and it is presented by COST. In the big step greedy search algorithm, a new polytope will only be accepted to replace the old or previous one when it has fewer or equal number of neo-epitopes.After the specific iteration (Itera), the optimized polytope O1 is produced. The preferred linkers input is optional, and only the greedy search will be applied when the preferred linkers are provided. [10–12].

The pseudo-algorithm for applying the big step greedy search in our work is presented below:

Algorithm 1

Itera = number of ***BM*** to attempt;
E1 = Generate a polytope vaccine by ***BM***;
For i=1 to Itera:
    E2 = Generate a polytope vaccine by ***BM***.
    ΔE = ***COST*** (E2) - ***COST*** (E1)
    If (ΔE <= 0):
        E1 = E2
/* end for loop*/
O1 = Optimized vaccine E1

2.4 Monte carlo simulated annealing approach

The goal of optimizing polytope vaccine is to try to have 0 neo-epitope which is referred to as a global minimization problem. Compared to the greedy search which usually ends up in a local minimum near the starting point, monte carlo simulated annealing approach has chance to get out of the local minima because of the cooling schedule.
The pseudo-algorithm for applying the monte carlo in our work is presented below:

Algorithm 2

initT = initial temperature;
miniT = lowest temperature;

eta = cooldown factor;
t = current temperature;
itera = number of ***SM*** to attempt at t;
E1 = ***O1***;
While t > miniT:
    For i=1 to itera:
        E2 = Generate a polytope vaccine by ***SM***.
        ΔE = ***COST*** (E2) - ***COST*** (E1)
        If (ΔE <= 0) or exp(-ΔE/T) > rand(0,1):
            E1 = E2
    /* end for loop*/
    t = t - eta
O2 = Optimal vaccine E1

Some parameters are needed to be initialized, such as the initial temperature (initT), lowest temperature (miniT), cooldown factor (eta) and the number (itera) of small moves (***SM***) to attempt at each temperature. The function ***SM*** is used to produce a new polytope vaccine by making a small change on the previous vaccine. Four different types of small moves are defined by us, including exchanging the random two selected epitopes, deleting or adding or changing one amino acid within one of the linkers or spacers for the polytope vaccine. For each movement, a random one of the four functions will be chosen to produce a new polytope vaccine by applying ***SM***.
The cost function (***COST***) in our work relates to the number of neo-epitopes and the length of the linkers for the polytope vaccine. For each polytope vaccine (v), the cost function ***COST***(v) is shown in Eq. 1, where *No.neo* is the number of the neo-epitopes that can bind strongly with the MHC-I alleles given by NetMHCpan-4.1 prediction, L_l is the l^{th} linker in the vaccine and the *thre* is the longest length of the linkers the user wants. λ is the weight the user will penalize the linkers that are longer than *thre*.

$$\boldsymbol{COST}(v) = No.neo + \lambda \sum_{l} (L_l - thre) \quad (1)$$

As we known, the control parameter that gives chance to get out of local minima is the current temperature (t) from ***Algorithm 2***. The random small move that gets to lower energy is accepted but to higher energy is accepted with probability (***P***), where the $\Delta \boldsymbol{E}$ is the energy increase and t is the current temperature shown in the Eq. 2.

$$\boldsymbol{P} = exp(-\Delta \boldsymbol{E}/t) \quad (2)$$

The annealing procedure is melting the system at a high temperature so the acceptance ratio at the beginning is quite high, then gradually lowering the temperature based on the cooling schedule and taking proper small moves at each temperature until the system froze where the acceptance ratio is low[12–15]. In our work, the starting point of ***Algorithm 2*** is the optimized polytope vaccine ***O1*** given by ***Algorithm 1***. The optimal vaccine in our whole procedure is ***O2***.

3 Results

3.1 The optimal polytope vaccine with 0 neo-epitope is achievable

At the beginning, the λ from Eq. 1 is set to 0, so only the number of neo-epitopes of the vaccine is considered for the cost function. We successfully constructed the polytope vaccines with 0 neo-epitope at last for hepatitis C virus (HCV), coronavirus disease 2019 (COVID-19), influenza viruses (Flu), smallpox and respiratory syncytial virus (RSV). From Fig 2, all ***I*** are the initial polytope vaccines and ***O1*** are the optimized vaccines from the greedy search (***Algorithm 1***), and all ***O2*** are the optimal vaccines from monte carlo simuated annealing approach (***Algorithm 2***). The lower case presents the linkers and the uppercase stands for the selected epitopes. **A** is for Flu where ***I*** has 7 neo-epitopes, those are IILNMLSTV, PFEKSTVM, MFGILGFVF, AFMYSDFHF, EDELRSRYW, and IEDELRSRY which is shown up twice meaning it is presented on two different alleles. ***O1*** gave us 3 neo-epitopes including KFMYSDFHF, MYSDFHFII, and YSDFHFIIK. **B** is for RSV where the initial one ***I*** has 14 neo-epitopes, those are HVRVFYLEK, KYTHVRVFY, ASLLSLFGR, SSSKYTHVR, CPDIPYSGL, EVRQMLAEL, LQNPKASLL,

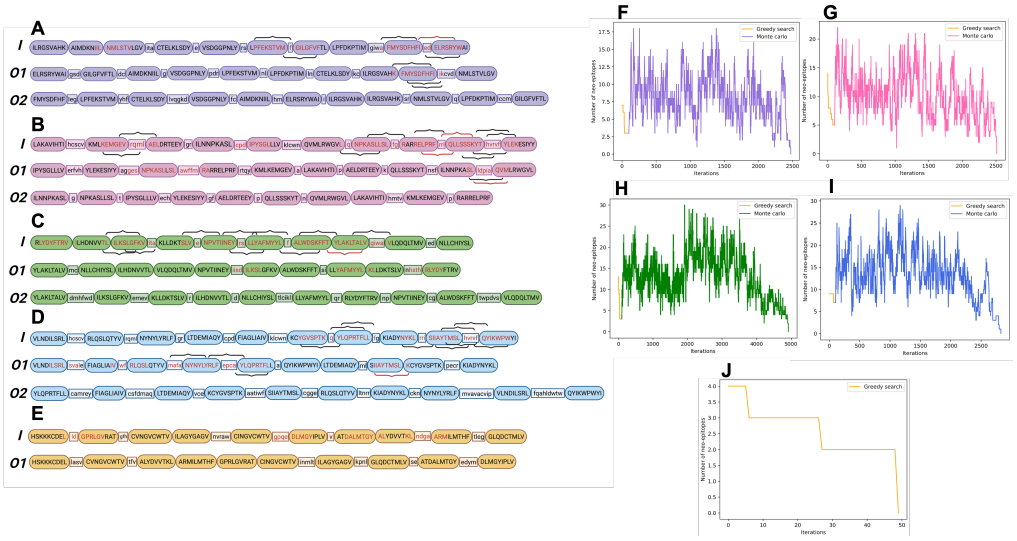

Figure 2: *The designed polytope vaccines created with Biorender.com without considering the length of the linkers*
The 5 polytope vaccines constructed for Flu (A and F), RSV (B and G), Smallpox (C and H), COVID-19 (D and I) and HCV (E and J). Each one has 2 or 3 states that shown from A to E, where ***I*** is the initial state and ***O1*** is the optimized one from Algorithm 1, and ***O2*** is the optimal vaccine from Algorithm 2. The red letter and curly brace are used to present the neo-epitope(red curly brace means the neo-epitope is presented twice). And the number of the neo-epitopes(y axis) for each iteration(x axis) are shown from F to J.

YTHVRVFYL, KEMGEVRQM, RELPRFRRI, RIQLLSSSK and LPRFRRIQL where the last two epitopes are presented twice respectively. ***O1*** from **B** contains 5 neo-epitopes including SLAWFFMRA, SLLDPIAQV, GESNPKASL, and LLDPIAQVM that is presented twice. **C** is for Smallpox where the ***I*** has 13 neo-epitopes, those are SLGFKVITA, SLVENPVTI, INEYRSLLY, LWDSKFFTY, LILKSLGFK, LYDYFTRVI, EYRSLLYAF, LYAFMYYLF, LFALWDSKF, TALVGIWAV, TLILKSLGF, and TYLAKLTAL that is presented twice. ***O1*** from **C** contains 3 neo-epitopes, including IISDILKSL, HSTHRLYDY, YAFMYYLKL. **D** is for COVID-19, ***I*** has 9 neo-epitopes, ***O1*** contains 7 neo-epitopes. **E** is for HCV, ***I*** has 4 neo-epitopes, ***O1*** gave us 0 neo-epitope which means ***Algorithm 2*** doesn't need to be applied. All ***O2*** have 0 neo-epitope, and thus we successfully constructed the polytope vaccines for the 5 viruses.

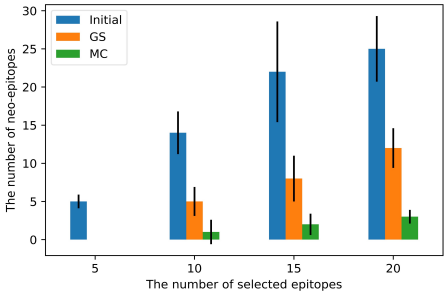

Figure 3: *Comparison the number of neo-epitopes with different sizes and states*
The blue bar stands for ***I***. The orange bar stands for ***O1*** from the greedy search (GS), and the green bar stands for ***O2*** from the monte carlo simulated annealing approach (MC).

After producing the 5 polytope vaccines successfully, we mixed the 50 epitopes that 10 from each into a soup. And the sets of size {5, 10, 15, 20} is chosen where the size means the number of randomly selected epitopes. For each size, our whole procedure is repeated for

8 times in total with the iteration equals to 100 for ***Algorithm 1*** repeated 5 times within one whole procedure, the finite steps equals to 30 at each iteration and the cooldown factor equals to 0.11 for the ***Algorithm 2***. Fig 3 is the error bar showing uncertainty of the number of neo-epitopes for different sizes based on calculating the standard deviation. The average number of neo-epitopes of ***I*** for the size 5, 10, 15, and 20 is 5, 14, 22 and 25. The average number of neo-epitopes of ***O1*** for the size 5, 10, 15, and 20 is 0, 5, 8, and 12. The average number of neo-epitopes of ***O2*** for the size 10, 15, and 20 is 1, 2, and 3. With the increasing number of selected epitopes, the difficulty of hitting 0 neo-epitope will increase. However, By modifying the *itera* and *eta*, such as keeping increasing the *itera* and decreasing the *eta*, 0 neo-epitope is achievable in theoretically.

3.2 The running time analysis

The running time for constructing the 5 polytope vaccines from Fig 2 is ranging from 20 minutes to more than 5 hours. For the repeated experiment shown in Fig 3, the running time was evaluated and shown in Fig 4. The running time for constructing the polytope vaccine with 5 selected epitopes is quite short (< 5mins), while the average running time for the vaccine with 20 selected epitopes is more than 10 hours. The reason for the increasing time with the same finite steps (30) at each temperature and cooldown factor (0.11) for the increasing size is the time increased for running NetMHCpan-4.1 to predict the binding affinity between the peptides and the MHC alleles and evaluating the cost function. The running time is the obvious limitation of our work.

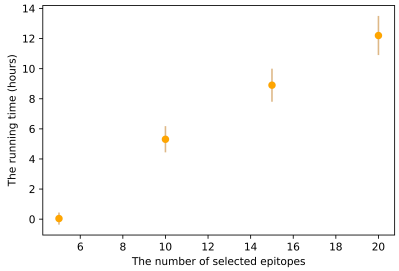

Figure 4: *Running time analysis for different sizes*

3.3 The cost for the length of the linkers

The 0 neo-epitope is achievable and has been proved for constructing polytope vaccines without considering the length of the linkers. However, the longer the linker, the higher risk the vaccine arises the unwanted or unexpected immune response. Thus, the cost function from ***Algorithm 2*** actually includes the penalty based on the specific weighting parameter λ for the length of the linkers that is longer than the threshold the user set which is *thre* from Eq. 1. What we tried is setting the *thre* equals to 6 and the λ equals to 1 for constructing the polytope vaccine for COVID-19 due to it has the linker longer than 6 amino acids compared with the other 4 vaccines. 3, 2, 2, 2, and 2 neo-epitopes were obtained for having 30, 35, 40, 45 and 55 finite steps at each temperature. The goal of having 0 neo-epitope for the polytope vaccine is harder to achieve if considering the cost function for the length of the linkers.

3.4 Webserver

The webserver PolyVaccine-1.0 is build for non-commercial use at `https://services.healthtech.dtu.dk/services/NetPolyEV-1.0/`.

3.4.1 Submission page

Three main fields for the input data are included: the selected epitopes either should be filled in the box or be uploaded from the user's

local disk, the selected MHC class I alleles, and the preferred linkers which is optional. The selected epitopes must be matching the selected MHC class I alleles to ensure our webserver or program run successfully. The preferred linkers should be shorter than 10 amino acids long for each. Only ***Algorithm 1*** will be implemented if the preferred linkers are provided. Additionally, the additional configuration is provided for the user to define the cost function for the length of the linkers where the weighting for the cost (linker) stands for λ and the threshold for the length of linkers stands for *thre* in ***Algorithm 2***. To keep optimizing the vaccine given by the big step greedy search. The slow monte carlo method can be applied afterward and there are three kinds of iteration which are the finite steps at each temperature that can be chosen, those are 30, 40, and 50. At most 10 selected epitopes and MHC alleles per submission considering the running time, and each epitope does not have more than 14 amino acids and not less than 8 amino acids. If the user wants to construct a polytope vaccine with more than 10 selected epitopes and alleles, the python file of our program is recommended to be downloaded.

3.4.2 Output page

After submitting the job, the first output page provides the information about the optimized vaccine ***O1***, the number of neo-epitopes of it and the table for the presentation of selected epitopes. If the number of neo-epitopes from ***O1*** is greater than 0, then the first output page will include the table for the presentation of neo-epitopes and the *itera* from ***Algorithm 2*** that can be selected (30, 40, or 50) by the user. To applying the monte carlo simulated annealing approach, the user must click the button 'Implement with slow Monte Carlo method'. At last, the optimal vaccine with its number of neo-epitopes and the tables for selected epitopes and neo-epitopes(if has) will be obtained.

4 Discussion

In this work, we propose to mainly apply the monte carlo simulated annealing approach in python for designing the optimal CD8+ T cell epitope-based vaccine with optimal epitope order and optimal linkers with flexible length and sequence. The goal of constructing a polytope vaccine is to elicit a potentially strong T-cell immune reaction by selecting epitopes that match the alleles and to avoid neo-immunogenicity by having as few as neo-epitopes as possible. NetMHCpan-4.1 is used to predict the binding affinity of peptides-HLA class I alleles and then the cost function for a specific polytope vaccine can be evaluated. The efficacy of the program was shown that the 0 neo-epitope of the vaccine can be achieved. With the increasing number of the selected epitopes, both the difficulty of hitting 0 neo-epitopes and the running time are increased. If the cost function for the length of linkers is included, 0 neo-epitope will be harder to have. Although the 0 neo-epitope is proved to be achievable in our work. An obvious limitation of the current method is it heavily relies on the cooldown factor and the number of iterations at each temperature from the monte carlo simulated annealing approach: a very slow cooldown factor and a high number of iterations will waste unnecessary computer time, conversely, a rapid cooldown factor and a low number of iterations will freeze the system quickly to get trapped in local minima and thus 0 neo-epitope cannot be obtained. This is the reason we stop at predicting CD8 + T cell epitopes with 9 amino acids long.

References

(1) Toussaint, N. C.; Maman, Y.; Kohlbacher, O.; Louzoun, Y. Universal peptide vaccines–optimal peptide vaccine design based on viral sequence conservation. *Vaccine* **2011**, *29*, 8745–8753.

(2) Schubert, B.; Kohlbacher, O. Designing string-of-beads vaccines with optimal spacers. *Genome medicine* **2016**, *8*, 1–10.

(3) Schubert, B.; Brachvogel, H.-P.; Jürges, C.; Kohlbacher, O. EpiToolKit—a web-based workbench for vaccine design. *Bioinformatics* **2015**, *31*, 2211–2213.

(4) Yang, B.; Hahn, Y. S.; Hahn, C. S.; Braciale, T. J. The requirement for proteasome activity class I major histocompatibility complex antigen presentation is dictated by the length of preprocessed antigen. *The Journal of experimental medicine* **1996**, *183*, 1545–1552.

(5) Dorigatti, E.; Schubert, B. Joint epitope selection and spacer design for string-of-beads vaccines. *Bioinformatics* **2020**, *36*, i643–i650.

(6) Jurtz, V.; Paul, S.; Andreatta, M.; Marcatili, P.; Peters, B.; Nielsen, M. NetMHCpan-4.0: improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. *The Journal of Immunology* **2017**, *199*, 3360–3368.

(7) Reynisson, B.; Alvarez, B.; Paul, S.; Peters, B.; Nielsen, M. 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 research* **2020**, *48*, W449–W454.

(8) Gonzalez-Galarza, F. F.; McCabe, A.; Santos, E. J. M. d.; Jones, J.; Takeshita, L.; Ortega-Rivera, N. D.; Cid-Pavon, G. M. D.; Ramsbottom, K.; Ghattaoraya, G.; Alfirevic, A., et al. Allele frequency net database (AFND) 2020 update: gold-standard data classification, open access genotype data and new query tools. *Nucleic acids research* **2020**, *48*, D783–D788.

(9) Fleri, W.; Paul, S.; Dhanda, S. K.; Mahajan, S.; Xu, X.; Peters, B.; Sette, A. The immune epitope database and analysis resource in epitope discovery and synthetic vaccine design. *Frontiers in immunology* **2017**, *8*, 278.

(10) Chandu, D. P. Big step greedy heuristic for maximum coverage problem. *International Journal of Computer Applications* **2015**, *125*.

(11) Dechter, A.; Dechter, R. On the greedy solution of ordering problems. *ORSA Journal on Computing* **1989**, *1*, 181–189.

(12) Rutenbar, R. A. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices magazine* **1989**, *5*, 19–26.

(13) Nourani, Y.; Andresen, B. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General* **1998**, *31*, 8373.

(14) Andresen, B.; Hoffmann, K. H.; Mosegaard, K.; Nulton, J.; Pedersen, J. M.; Salamon, P. On lumped models for thermodynamic properties of simulated annealing problems. *Journal de Physique* **1988**, *49*, 1485–1492.

(15) Nayeem, A.; Vila, J.; Scheraga, H. A. A comparative study of the simulated-annealing and Monte Carlo-with-minimization approaches to the minimum-energy structures of polypeptides:[Met]-enkephalin. *Journal of Computational Chemistry* **1991**, *12*, 594–605.