Computational prediction of *Trypanosoma cruzi* epitopes towards the generation of an epitope-based vaccine against Chagas disease

Albert Ros-Lucas[1,2*], David Rioja-Soto[1], Joaquim Gascón[1,2], Julio Alonso-Padilla[1,2*]

[1]Barcelona Institute for Global Health (ISGlobal), Hospital Clinic - University of Barcelona, Barcelona, Spain

[2]CIBERINFEC, ISCIII—CIBER de Enfermedades Infecciosas, Instituto de Salud Carlos III, Madrid, Spain

* Correspondence: ARL albert.ros@isglobal.org; JAP julio.a.padilla@isglobal.org

Abstract

Chagas disease, caused by the protozoan parasite *Trypanosoma cruzi*, is considered a Neglected Tropical Disease. Limited investment is assigned to its study and control, even though it is one of the most prevalent parasitic infections worldwide. An innovative vaccination strategy involving an epitope-based vaccine that displays multiple immune determinants originating from different antigens could counteract the high biological complexity of the parasite and lead to a wide and protective immune response. Here we describe a computational reverse vaccinology pipeline applied to identify the most promising peptide sequences from *T. cruzi* proteins, prioritizing evolutionary conserved sequences, to finally select a list of T and B-cell epitope candidates to be further tested in an experimental setting.

Key words: Chagas disease, *Trypanosoma cruzi*, Epitopes, Reverse Vaccinology,

1. Introduction

Chagas disease affects about 7 million people worldwide, but it is in the Americas where it exerts its highest burden [1]. Even though it is the third most common parasitic infection, it remains neglected [2], and very little resources are invested to its study and control. Moreover, in the last few decades the disease has spread its impact and become emerging in Europe, the USA and other non-endemic regions linked to population movements [3]. The capability of *Trypanosoma cruzi*, que causative agent, to express diverse antigens that vary among its distinct life cycle stages, as well as products that interfere with immune response, allows it to evade antigen presentation and host defense mechanisms [2]. There are currently two drugs to treat the infection: benznidazole and nifurtimox. Both are highly efficacious against the short (8-12 weeks long) acute stage of the disease, which is asymptomatic and generally goes undiagnosed and therefore untreated. However, it is in the chronic stage (years or even decades long), when the characteristic heart and digestive symptomatology appears, that the infection gets diagnosed and drugs administered to the patient. Unfortunately, by then both are not that efficacious anymore. Besides, their administration leads to frequent toxicity, which can even lead to treatment discontinuation [4], [5]. Moreover, in many cases diagnosis and treatment arrive too late, when the patient is already suffering from incapacitating heart and/or gut tissue disruptions [2].

In view of the existing difficulties for patients to be both diagnosed and treated before it is too late to halt the infection, the development of an effective vaccine would mean a breakthrough. Several approaches to generate a Chagas vaccine are underway based on attenuated *T. cruzi* strains, DNA technology or recombinant proteins, but most of them rely on the use of a single or a handful antigens or their fragments [6]. In contrast, an epitope-based approach in which several antigens could be represented at the same time in a unique construct would represent a promising alternative. Both B-cell epitopes and $CD8^+$ and $CD4^+$ T-cell epitopes would be included in this construct. With that insight into account, the development of a computational pipeline is needed in order to explore the parasite protein space and narrow down the peptides that could reunite all the desired characteristics that would make them relevant immunogenic epitopes, able to trigger a functional immune response against *T. cruzi* infection. The Immune Epitope Database (IEDB) and its Analysis Resource [7] is a publicly available resource which hosts tools to help in the prediction of epitopes. While generation, processing and development of T-cell epitopes, especially $CD8^+$, are well-known,

in case of B-cell epitopes there are still some unclear areas to overcome[8]. Nevertheless, parasite-specific B-cell responses have been described as crucial for maintaining a systemic response and prevent exhaustion of parasite-specific $CD8^+$ T cells [9], plus parasite-specific antibodies are fundamental to stunt the infection of new cells by free-swimming trypomastigotes [10]. Given the importance of protein residues arrangement in 3D for the recognition of epitopes by antibodies, structural information is fundamental to improve the prediction of B-cell epitopes. However, *T. cruzi* is an understudied pathogen and protein structures of the parasite are scarce. To compensate for this, we will rely on the AlphaFold Protein Structure Database [11] for the prediction of B-cell epitopes.

2. Methods

2.1 Collection of pathogenic *T. cruzi* proteomes

1. Open TriTrypDB at http://tritrypdb.org [12]. This site, part of the Eukaryotic Pathogen, Vector and Host Informatics Resource (VEuPathDB) [13] suite of databases, provides access to genome-scale datasets for kinetoplastid parasites, as well as several research and analysis tools.

2. In the main menu, go to Data > Download data files > Current_release and open one by one the genomes' folders you are interested in (see Note 1). Known non-pathogenic strains, such as *T. cruzi* marinkellei, can be excluded from the analysis. For each of them, open the "fasta/data/" folder and download the "AnnotatedProteins.fasta" file. In case no annotated protein file is available for this organism, you can download the ORF.gff file in the "gff/data/" folder. These .gff files contain the translation product of the ORF, which is what we are interested in. Finally, choose a reference strain that is well annotated, that is, one that has an annotated proteins file. The reference strain for *T. cruzi* in TriTryp is CL Brener Esmeraldo-like, which will be the one used in this protocol.

3. For each downloaded strain, choose either the annotated proteins fasta file, or the ORFs gff file. The annotated proteins file can be used as is, but it's preferable to modify the headers so that all strains have some consistency. The gff files must be parsed first, extracting the translation products and formatting them as fasta entries (see Notes 2-3). Finally, merge all the sequences into a single fasta file that contains all the selected strains.

2.2. Clusterization and filtering

1. In order to group all the different sequences, a clustering software is used. In this protocol we will use CD-HIT [14], which is very fast and easy to use. It can be downloaded here https://github.com/weizhongli/cdhit/releases, or installed via a package management system such as conda (see Note 4). Launch the program using the following command:

```
cd-hit -I input.fasta -o tcruzi -c 0.9 -n 5 -d 0 -g 1 -M 0 -T 0
```
,

`-i` specifies the path to the fasta file, `-o` the path and name of the output files, `-c` the threshold of sequence identity (90% by default), and `-n` the word length (default 5) (see Note 5). In addition, `-d 0` specifies that the full header (up to the first space) will appear in the output, `-g 1` tells the program to cluster any sequence to the most similar cluster (slower, but more accurate), and `-M 0` and `-T 0` allows CD-HIT to use unlimited memory and CPUs, respectively (see Notes 6-7).

2. CD-HIT provides two outputs, we will use the *.clstr* file which lists the clusters and the sequences in them. This file can be parsed using a programming language, or you can use the *make_multi_seq.pl* Perl script included with CD-HIT to generate a fasta file for each cluster (be careful, as this might generate tens of thousands fasta files, depending on the identity threshold used and the number of sequences) (see Note 8).

3. In order to keep only the clusters we are interested in, several criteria will be used:
   - Only keep clusters with at least one sequence from our reference strain (in this case, CL Brener Esmeraldo-like).
   - Only keep clusters with a minimal representation of the strain diversity. For example, if we are using 20 strains, keep those clusters with sequences from at least 10 (less restrictive) or 15 (more restrictive) different strains.
   - When predicting B-cell epitopes, additional filtering is needed. This will be explained further below.

2.3 Alignment and conservation analysis

1. The sequences of each cluster must be aligned. For this, you can use your multiple sequence alignment algorithm of choice. For this protocol, we will use MUSCLE [15]. It can be downloaded from here https://drive5.com/muscle/downloads_v3.htm or by using a software package manager. For each cluster, launch MUSCLE by using the following command (see Note 9):

```
/path/to/muscle -align cluster_1.fasta -output cluster_1_aln.fasta
```

2. For each cluster, now aligned, a reference sequence must be designated. One way to do so is to generate a consensus sequence for that alignment, for example by creating a sequence from the most common residue in each position. Then, score each sequence in the alignment against this consensus sequence. For this you can proceed with a global pairwise alignment, selecting the sequence most similar to the consensus as reference (see Note 10). Another simpler option is to choose an aligned sequence from the reference strain. In case of draws, different priorities can be given to sort out the most relevant sequence.

In order to assess conservation, we will calculate the Shannon entropy of each position [16], [17]. This will quantify the amount of variation contained in a specific position among all the cluster members. It consequently gives information about how well conserved is each position of the cluster, so that we are able to discard those regions that are not conserved enough. The following equation is used:

$$H = -\sum_{i}^{M} P_i log_2(P_i),$$

where M is 20 (total number of amino acids) and P_i is the frequency of a given amino acid i in each position. H can range from 0 (if there is no variability in a specific position) and 4.32 (when all the 20 amino acids are equally represented in a position) [18] (see Note 11).

3. Choose an entropy threshold above which positions will be flagged as non-conserved. A threshold of 0 will ensure that all positions are totally conserved among the sequences in an alignment. However, this might cause situations in larger alignments in which if even one residue is different, a position will be discarded. A more sensible approach is to use thresholds of 0.5 or 1. You can tag the non-conserved positions with an asterisk for future reference (see Notes 12-13).

2.4 CD8$^+$ T-cell epitope prediction

1. The first step is to predict proteasomal processing, for which we will use NetChop [19]. It can be found at the Technical University of Denmark (DTU) department of Health Technology page at https://services.healthtech.dtu.dk/service.php?NetChop-3.1. In order to correctly assess the cleavage sites, the unaligned reference sequence from each cluster should be used (see Note 14). The default C-term 3.0 prediction method and 0.5 threshold are recommended. You can either run it using the long or the short output method, but parsing will be different in each case. In both cases, an S in a position indicates that the bond between that residue and the next is cleaved. Use this output, plus the results of the entropy calculation, to split all the input sequences (Figure 1).

2. Since CD8$^+$ T-cell are predominantly 9-mers, cleaved fragments shorter than that can be discarded. Any k-mer larger than 9 residues must be converted to its constituting 9-mers (see Note 15). For example, the 10-mer `REDSHIRTTS` has to be converted into `REDSHIRTT-` and `-EDSHIRTTS` (without hyphens). Finally, merge all 9-mers into a single file, one peptide per line (see Note 16).

3. Peptides are imported into the ER via the transporter associated with antigen processing (TAP). To predict binding to this transporter, we will use a TAP binding prediction tool created by Besser and Louzoun [20]. Go to https://transition-probability.shinyapps.io/TAP_binding_App/ and upload the file that contains one 9-mer per line. The output is a CSV file with two columns: the first is a probability of binding to TAP, and the second is an estimate of the $\log(IC_{50})$. We will use the latter to discern which peptides are more likely to bind, selecting those with a $\log(IC_{50})$ below 1 (see Note 17).

4. Finally, we will predict the binding affinity to the MHC I. The current recommended method by IEDB is NetMHCpan EL 4.1 [21], which can be downloaded either from DTU or from the same IEDB. The latter contains additional prediction methods, and results are easier to parse. Download it from here http://tools.iedb.org/mhci/download/, and install it following the straightforward readme

instructions. To launch it, you will need a fasta file with all 9-mers (see Note 18), and a list of HLA class I alleles. It is recommended to use the IEDB reference set of 27 alleles found here https://help.iedb.org/hc/en-us/articles/114094151851, which have maximal population coverage [22] (see note 19).

5. The basic command to launch a prediction, assuming the current working directory is the main IEDB installation, is the following:

```
src/predict_binding.py IEDB_recommended HLA-A*01:01 9 input.fasta > results.txt
```

This will launch the prediction for all peptides inside *input.fasta* using the HLA-A*01:01 allele with a peptide length of 9, and saving the results into *results.txt* (see Note 20).

6. The result is a ranking of each peptide to a specific HLA allele. Depending on how you launched the prediction, you will have either 27 files with one allele per file, or one single file with all results. The results file is structured as tab-separated values, so you can parse it directly in Excel. The most important is the "rank" column, which compares the score of a specific peptide to a set of random peptides. Peptides with a rank below 1% are considered strong binders, so we will keep them (see Note 21).

2.5 CD4+ T-cell epitope prediction

1. In contrast to MHC I, there is no accurate predictor of endosomal protease processing. In this case, we proceed directly with the conserved sequences. Since typically $CD4^+$ T-cell epitopes are 15-mers, select all conserved fragments (those with entropies below threshold) with a length of 15 or more residues. Then, download the IEDB MHC II epitope predictor here http://tools.iedb.org/mhcii/download/ and install it following the readme instructions. Additionally, download the HLA class II reference set [23] from https://help.iedb.org/hc/en-us/articles/114094151851.

2. Similar to $CD8^+$ epitopes, predictions for each allele can be launched separately or in a single command (see Note 22):

```
python mhc_II_binding.py IEDB_recommended HLA-DRB1*03:01 input.fasta > results.txt
```

This tool will take any k-mer bigger than 15 residues and assess all its constituent 15-mers individually. The percentile rank column can be used to select the best binders; for MHC II peptides below the 10% are considered good binders, but wide allelic coverage is more important. Choose those epitopes in the top 10% that match two (or more) different alleles (see Note 23).

2.6 B-cell epitope prediction

B-cell epitopes are recognized directly by B lymphocytes, and as such, they need to be accessible. That means that antigens that contain these epitopes must be located in the surface of the pathogen. In *T. cruzi*, proteins in the membrane are mainly anchored either by transmembrane domains, or by GPI anchors.

1. Download the most recent version of SignalP [24] from here https://services.healthtech.dtu.dk/service.php?SignalP (currently 6.0), which predicts which proteins contain signal peptides and are thus selected for the secretory pathway. Install the SignalP package following the readme instructions (see Note 24).

2. Using the annotated proteins fasta file from the reference strain, launch SignalP:

```
signalp6 -fasta input.fasta -output_dir ref_signalp -org eukarya -fmt none
```

This will create a folder in your current working directory called *ref_signalp*, and all the output will be found inside. The most important is the *prediction_results.txt* file. This is a tab-separated values text file with the predictions for each protein. The "Prediction" column determines if the protein has a signal peptide (SP) or not (OTHER). Discard any proteins that fall in the latter category.

3. With the subset of proteins with a signal peptide, create a new fasta file containing only these and go to the NetGPI [25] site https://services.healthtech.dtu.dk/service.php?NetGPI-1.1. Since this software is not currently available for local installation, upload your fasta file and run the prediction online

using the “short” output method (see Note 25). Click on the “Prediction summary”, which will export a tab-separated values file that categorizes all proteins into either “GPI-Anchored” or “Not GPI-Anchored”.

4. Create two new files, one that contains all the gene IDs, one per line, associated with a signal peptide (result of SignalP), and another that contains all the IDs that are GPI-anchored.

5. We will now create a new search strategy at TriTrypDB that will allow us to select those proteins that are putatively exposed in *T. cruzi*. In the site home, on the left menu, search and click in “GO Term” to identify genes based on gene ontology terms. In the new page, select only the reference strain CL Brener Esmeraldo-like, and in the GO Term box input “cell surface” or “GO:0009986”. The other parameters can be left as default.

6. We will filter out any proteins that are marked to go into the secretory pathway (those with a signal peptide), but that do not have any transmembrane domain. Add a new step into the strategy, and combine its results using the “1 minus 2” method (Figure 2). Using the “List of IDs” search, upload the text file with the IDs from SignalP. After applying this step, click on the small “edit” button in the upper left corner of the “IDs List” step, and select “Make nested strategy” (see Note 26). This useful feature allows us to compartmentalize different steps of our search strategy. Add a new step into the newly created nested strategy, and intersect the results of the IDs List search with a Transmembrane Domain Count search. Again, select the reference strain, and write 0 in both the minimum and maximum number of transmembrane domains.

7. Add a new step, this time unionizing the results from the step before. Upload again the list of IDs with signal peptide using the List of IDs search. Now, create a new nested strategy from this last step, intersecting these results with a new Transmembrane Domain Count search selecting those that have a minimum of one transmembrane domain. Finally, add a new step in the main search strategy, unionizing the results with an IDs List search in which we will upload the list of genes with a GPI

anchor.

8. Some of the selected genes, which we have predicted to locate in the cell surface, might instead be anchored in intracellular organelles' membranes. We will filter these out by adding a new "Go Term" search with the "1 minus 2" method, and searching for the GO term "intracellular organelle" or "GO:0043229" (see Note 27). You should now have a search strategy with a list of filtered genes below (Figure 3).

9. Click on the Download button above the genes list. From the pre-configured tables report, select the Links outs > External links table and download it as a tab- or coma-delimited file. From this file, keep only the external IDs from The Universal Protein Resource (UniProt), which will be useful later. This way, you will have a final list of genes, with their associated UniProt ID.

10. In the case of *T. cruzi*, experimentally determined protein structures are scarce. To circumvent this, we will use the models from the AlphaFold Protein Structure Database. The whole CL Brener strain proteome is available at https://alphafold.ebi.ac.uk/download. Using the list of external UniProt IDs, keep only the protein structures PDB files we are interested in. We will also discard any AlphaFold models that do not meet a quality threshold. Prediction accuracy in these models is measured as the predicted Local Distance Difference Test score (pLDDT) [26], with higher values representing better accuracy. In the PDB file format, this score is stored in place of the B-value. Using a PDB structure parser we will calculate the mean pLDDT of all residues (or alpha carbons) for each structure (see Note 28). If any structure has a mean pLDDT below 70, which is described as "confident" by the AlphaFold creators [27], we will discard it and its associated gene.

11. Return to the clustering results from CD-HIT. After filtering out the clusters as described before, we will also keep only those clusters that have at least one member pertaining to the surface genes list. The multiple sequence alignment and entropy calculations are done as described above. Finally, create a new fasta file with the unaligned reference sequences from each cluster.

12. To predict B-cell epitopes, we will use the BepiPred 2.0 predictor [28]. The installation of this software, however, is more complex compared to the T-cell predictors and will need additional software (see Note 29). Download BepiPred bundled in the IEDB B-cell epitope prediction tool http://tools.iedb.org/bcell/download/. You will also need to install NetSurfP [29] https://services.healthtech.dtu.dk/service.php?NetSurfP-3.0, specifically the 1.0d version (see Note 30), and a blastpgp installation from the unsupported BLAST [30] suite version 2.2.18 that you can download from here https://ftp.ncbi.nlm.nih.gov/blast/executables/legacy.NOTSUPPORTED/2.2.18/blast-2.2.18-x64-linux.tar.gz.

13. Using the unaligned reference sequences fasta, launch BepiPred-2.0:

    ```
    python predict_antibody_epitope.py -m Bepipred-2.0 -f input.fasta > results.txt
    ```

    BepiPred scores each position with the propensity of being part of a B-cell epitope. By default, the threshold is set at 0.5, so any position with a score above that will be considered as part of an epitope.

14. To make sure that the predicted epitopes are indeed located in the surface of a protein, we will calculate the accessibility of each residue. In this protocol we will use NACCESS [31] to calculate the relative solvent accessibility of each position. It only requires a PDB file as an input, for which we will use the *T. cruzi* AlphaFold models. To obtain and install NACCESS, follow the instructions on its site http://www.bioinf.manchester.ac.uk/naccess/ (see Notes 31-32). It is simply used by the command:

    ```
    naccess pdb_file.pdb
    ```

    This will output several files, but we are just interested in the *.rsa* results. Each residue will show several values in different columns, but we will only use the “All-atoms REL” column, which refers to the relative accessibilities in percentage of all the atoms in a residue. Residues with at least 50% RSA will be considered as good candidates of being part of B-cell epitopes.

15. Finally, we will compare the results of the entropy calculation (conservation), the RSA (accessibility) and BepiPred (epitope prediction). Since the entropy has been calculated from the multiple sequence alignments, while the RSA and BepiPred have been obtained from the unaligned sequences, we must now "realign" these results. You can either add gaps using the aligned reference sequence as model, or align the sequence into the MSA by using MUSCLE's profile-profile alignment (see Note 33). Finally, analyze every position of each sequence. Regions that fulfill all three requirements will be considered as potential B-cell epitopes. While B-cell epitopes do not have a canonical length, for practical purposes we will keep sequences with a minimum length of 7 amino acids.

2.7 Homology analysis

Given that epitope sequences similar to host proteins could trigger cross-reactivity or show low immunogenicity due to mechanisms of immunotolerance, it is useful to study sequence homology between potential epitopes and the proteome of *Homo sapiens*.

1. Launch a BLASTP [30] search of our predicted epitopes to assess the similarity between these sequences and the ones from the host. You can either run this search locally, or by using the NIH BLAST site. Blast against the human non-redundant (nr) protein sequences database with the following parameters: PAM30 matrix, word size of 2, gap open and extend costs of 9 and 1, no compositional adjustments, and an expect threshold of 10000.

2. If you have BLAST installed locally, we can also do a BLASTP search against the human microbiome, since it heavily influences the immune response. For this, we can create a new blast database using sequences from the Human Microbiome Project (HMP) [32], which can be accessed in the NIH Bioproject 43021 (https://www.ncbi.nlm.nih.gov/bioproject/43021) (see Note 34). Run the BLASTP search locally using this database and the same parameters as above.

3. An epitope's identity can be measured as the number of identical positions over the aligned query length. If any epitope finds a hit with an identity higher than 70% in either the human or the HMP blast searches, discard it.

3.Notes

1. A script can be used instead of downloading each strain manually. For this purpose, you can use a web scrapper with the https://tritrypdb.org/common/downloads/Current_Release/ base url, instead of the web app.
2. You can use the strain name and the gene id (i.e., >TcruziCLBrener_TcCLB.403481.10). It's important to not use spaces in the headers, so use underscores ("_") or similar to join both identifiers
3. For these files, the strain name and the locus tag can be used as headers (e.g., >TcruziBrazilA4_TcBrA4_Chr1-6-1073797-1073507).
4. You can install conda using the Anaconda o Miniconda installers. Anaconda is more user-friendly, but has a much heavier installation that contains many packages. Miniconda is used entirely via the command prompt, but is much more lightweight.
5. The identity cutoff and the word size can be set to lower thresholds, which will result in fewer clusters, but with more sequences in them. For a guide on which word size to use for each cutoff, consult the CD-HIT manual.
6. If the input fasta is too large and the program collapses your computer, try adjusting the `-M 0` and `-T 0` parameters to limit the resources consumed.
7. Additional parameters can be useful, and visualized with `cd-hit -h`. For example, `-s 0.75` specifies that the shorter sequences of a cluster need to be, at least, 75% of the length of the representative sequence in that cluster. This can be very useful if we have many protein fragments that we want to discard, which would distort the alignments otherwise.
8. This file is usually found in the installation folder of CD-HIT, and you can also find it in its GitHub repository. However, you will need to install Perl in your system.
9. Given the large number of files, some scripting will be necessary to align all the fasta files. Parallelization, if possible, is recommended.
10. The BioPython library for Python contains the pairwise2 module, which you can use to perform quick global pairwise alignments and return a score for each one.

11. The Shannon entropy equation can be implemented manually, or using an existing library or function for the programming language of choice. In Python, the SciPy library has an entropy function that can be used (remember to set the log to base 2).
12. Multiple sequence alignments with shorter proteins than the rest (e.g., short ORFs or other protein fragments) will cause strange results in the N and/or C-terminal regions. To avoid this, you can exclude these sequences when clustering using the `-s` parameter, or not taking into consideration the artificial gaps generated by these sequences in both termini. Normal gaps product of indels should be taken into consideration though.
13. The output from the entropy calculation can be stored as the entropy for each position (e.g., in a CSV file) or as an aligned sequence with masked positions. The former can be useful if you want to try different entropy thresholds.
14. Given that the allowed number of sequences for the online site is 100, it is better to install the software and run it locally with a single fasta file. Alternatively, you can split the sequences in several fasta files.
15. Since one protein can provide several k-mers, it is important to keep track of the origin of each of them. A good practice is to create a new fasta file with the newly generated 9-mers, adding underscores to the header of each one, keeping the original header information intact.
16. Keep the same order as in the fasta file generated beforehand.
17. These scores follow the same order as the input file, so they can be easily correlated with each 9-mer and thus their protein of origin.
18. It is possible that duplicated 9-mers from different (or the same) proteins appear. The MHC I predictor scores the same sequences equally, so if you have many repeated 9-mers you can predict the scores of just one of them.
19. Alternatively, you can use specific HLA alleles for a more regional approach. You can consult allele frequencies at http://www.allelefrequencies.net.
20. You can either launch 27 different commands, using a different allele each time (useful if you are interested in parallelizing execution), or launch a single command specifying all the alleles and lengths separated by commas. Remember that each allele must be matched by a specified epitope length:

```
./src/predict_binding.py    IEDB_recommended    HLA-A*01:01,HLA-A*02:01    9,9
input.fasta > results.txt
```

21. Some peptides might score better percentiles in some alleles, but not in others. Strong binders in several alleles are preferable, since they will theoretically be more widely recognized in the population.
22. Since the recommended method for each allele differs, it is better to specify `IEDB_recommended` as method.
23. Adjust the number of alleles needed to accommodate a reasonable number of $CD4^+$ T-cell epitopes. Remember that space in a genetic construct is limited, so strike a balance between the different epitope types and space. In addition, it is also recommended to also select $CD4^+$ epitopes that coincide with B-cell epitopes.
24. It is suggested to install SignalP in a separate Python environment, to avoid conflicts.
25. If there are too many entries and the prediction fails, split your fasta file accordingly.
26. For convenience, you can edit the names of every step and nested strategy.
27. Further filtering could be done by, for example, excluding genes that are up-regulated in epimastigote stages, and/or selecting those that show high evidence of expression in trypomastigote and amastigote stages.
28. BioPython can parse PDB files using the Bio.PDB.PDBParser function. Iterate through each atom of each residue, and check if they are alpha carbons ("CA") with the .get_id() method.
29. Follow the readme instructions of each package to install it; it is also highly recommended to create a new virtual environment just for BepiPred with Python 3.6 and the specific versions of the packages described in the BepiPred installation instructions.
30. NetSurfP-1.0d requires the data.tar.gz and nr70_db.tar.gz files to work, which can be downloaded from https://services.healthtech.dtu.dk/service.php?NetSurfP-1.0.
31. To obtain NACCESS, you will have to contact prof. Simon Hubbard and send a signed agreement via postal mail to The University of Manchester, in the United Kingdom. Since this can take time, take this into account when planning your project.
32. Some problems are known to occur in systems with modern Fortran compilers. Guidelines on how to solve this are included in the NACCESS readme instructions.

33. More information on profile-profile alignment can be found in the MUSCLE manual and/or here https://drive5.com/muscle/manual/addtomsa.html
34. Upon installing BLAST locally, you will also have access to the `makeblastdb` command. The easiest way to use it is to create a unique fasta file with all the sequences you want to include in this database. In the BioProject page, find the "Protein sequences" link, and then save all the sequences (about 7 million) using the Send to > File > Fasta option.

Acknowledgements

We acknowledge support from the Spanish Ministry of Science and Innovation and State Research Agency through the "Centro de Excelencia Severo Ochoa 2019-2023" Program (CEX2018-000806-S), and support from the Generalitat de Catalunya through the CERCA Program. This research was also supported by CIBER -Consorcio Centro de Investigación Biomédica en Red- (CB 2021), Instituto de Salud Carlos III, Ministerio de Ciencia e Innovación and Unión Europea—NextGenerationEU.

References

[1] C. M. Beaumier, P. M. Gillespie, U. Strych, T. Hayward, P. J. Hotez, and M. E. Bottazzi, "Status of vaccine research and development of vaccines for Chagas disease," *Vaccine*, vol. 34, no. 26, pp. 2996–3000, Jun. 2016, doi: 10.1016/j.vaccine.2016.03.074.

[2] M.-J. Pinazo and J. Gascon, "Chagas disease: from Latin America to the world," *RIP*, vol. 4, pp. 7–14, May 2015, doi: 10.2147/RIP.S57144.

[3] J. Gascon, C. Bern, and M.-J. Pinazo, "Chagas disease in Spain, the United States and other non-endemic countries," *Acta Tropica*, vol. 115, no. 1–2, pp. 22–27, Jul. 2010, doi: 10.1016/j.actatropica.2009.07.019.

[4] E. Aldasoro *et al.*, "What to expect and when: benznidazole toxicity in chronic Chagas' disease treatment," *Journal of Antimicrobial Chemotherapy*, vol. 73, no. 4, pp. 1060–1067, Apr. 2018, doi: 10.1093/jac/dkx516.

[5] Y. Jackson, E. Alirol, L. Getaz, H. Wolff, C. Combescure, and F. Chappuis, "Tolerance and Safety of Nifurtimox in Patients with Chronic Chagas Disease," *CLIN INFECT DIS*, vol. 51, no. 10, pp. e69–e75, Nov. 2010, doi: 10.1086/656917.

[6] C. Teh-Poot and E. Dumonteil, "Mining Trypanosoma cruzi Genome Sequences for Antigen Discovery and Vaccine Development," in *T. cruzi Infection*, vol. 1955, K. A. Gómez and C. A. Buscaglia, Eds. New York, NY: Springer New York, 2019, pp. 23–34. doi: 10.1007/978-1-4939-9148-8_2.

[7] R. Vita *et al.*, "The Immune Epitope Database (IEDB): 2018 update," *Nucleic Acids Res*, vol. 47, no. D1, pp. D339–D343, Jan. 2019, doi: 10.1093/nar/gky1006.

[8] K. A. Galanis, K. C. Nastou, N. C. Papandreou, G. N. Petichakis, D. G. Pigis, and V. A. Iconomidou, "Linear B-Cell Epitope Prediction for In Silico Vaccine Design: A Performance Review of Methods Available via Command-Line Interface," *IJMS*, vol. 22, no. 6, p. 3210, Mar. 2021, doi: 10.3390/ijms22063210.

[9] N. L. Sullivan, C. S. Eickhoff, J. Sagartz, and D. F. Hoft, "Deficiency of Antigen-Specific B Cells Results in Decreased *Trypanosoma cruzi* Systemic but Not Mucosal Immunity Due to CD8 T Cell Exhaustion," *J.I.*, vol. 194, no. 4, pp. 1806–1818, Feb. 2015, doi: 10.4049/jimmunol.1303163.

[10] A. Buschiazzo, R. Muiá, N. Larrieux, T. Pitcovsky, J. Mucci, and O. Campetella, "Trypanosoma cruzi trans-Sialidase in Complex with a Neutralizing Antibody: Structure/Function Studies towards the Rational Design of Inhibitors," *PLoS Pathog*, vol. 8, no. 1, p. e1002474, Jan. 2012, doi: 10.1371/journal.ppat.1002474.

[11] M. Varadi *et al.*, "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models," *Nucleic Acids Research*, vol. 50, no. D1, pp. D439–D444, Jan. 2022, doi: 10.1093/nar/gkab1061.

[12] M. Aslett *et al.*, "TriTrypDB: a functional genomic resource for the Trypanosomatidae," *Nucleic Acids Research*, vol. 38, no. suppl_1, pp. D457–D462, Jan. 2010, doi: 10.1093/nar/gkp851.

[13] B. Amos *et al.*, "VEuPathDB: the eukaryotic pathogen, vector and host bioinformatics resource center," *Nucleic Acids Research*, vol. 50, no. D1, pp. D898–D911, Jan. 2022, doi: 10.1093/nar/gkab929.

[14] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006, doi: 10.1093/bioinformatics/btl158.

[15] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792–1797, Mar. 2004, doi: 10.1093/nar/gkh340.

[16] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, Oct. 1948, doi: 10.1002/j.1538-7305.1948.tb00917.x.

[17] J. Alonso-Padilla, E. M. Lafuente, and P. A. Reche, "Computer-Aided Design of an Epitope-Based Vaccine against Epstein-Barr Virus," *Journal of Immunology Research*, vol. 2017, pp. 1–15, 2017, doi: 10.1155/2017/9363750.

[18] J. J. Stewart *et al.*, "A Shannon entropy analysis of immunoglobulin and T cell receptor," *Molecular Immunology*, vol. 34, no. 15, pp. 1067–1082, Oct. 1997, doi: 10.1016/S0161-5890(97)00130-2.

[19] M. Nielsen, C. Lundegaard, O. Lund, and C. Keşmir, "The role of the proteasome in generating cytotoxic T-cell epitopes: insights obtained from improved predictions of proteasomal cleavage," *Immunogenetics*, vol. 57, no. 1–2, pp. 33–41, Apr. 2005, doi: 10.1007/s00251-005-0781-7.

[20] H. Besser and Y. Louzoun, “Cross-modality deep learning-based prediction of TAP binding and naturally processed peptide,” *Immunogenetics*, vol. 70, no. 7, pp. 419–428, Jul. 2018, doi: 10.1007/s00251-018-1054-6.
[21] B. Reynisson, B. Alvarez, S. Paul, B. Peters, and M. Nielsen, “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*, vol. 48, no. W1, pp. W449–W454, Jul. 2020, doi: 10.1093/nar/gkaa379.
[22] D. Weiskopf *et al.*, “Comprehensive analysis of dengue virus-specific responses supports an HLA-linked protective role for CD8+ T cells,” *Proc Natl Acad Sci U S A*, vol. 110, no. 22, pp. E2046-2053, May 2013, doi: 10.1073/pnas.1305227110.
[23] J. Greenbaum, J. Sidney, J. Chung, C. Brander, B. Peters, and A. Sette, “Functional classification of class II human leukocyte antigen (HLA) molecules reveals seven different supertypes and a surprising degree of repertoire sharing across supertypes,” *Immunogenetics*, vol. 63, no. 6, pp. 325–335, Jun. 2011, doi: 10.1007/s00251-011-0513-0.
[24] F. Teufel *et al.*, “SignalP 6.0 predicts all five types of signal peptides using protein language models,” *Nat Biotechnol*, vol. 40, no. 7, pp. 1023–1025, Jul. 2022, doi: 10.1038/s41587-021-01156-3.
[25] M. H. Gíslason, H. Nielsen, J. J. Almagro Armenteros, and A. R. Johansen, “Prediction of GPI-anchored proteins with pointer neural networks,” *Current Research in Biotechnology*, vol. 3, pp. 6–13, 2021, doi: 10.1016/j.crbiot.2021.01.001.
[26] J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, doi: 10.1038/s41586-021-03819-2.
[27] K. Tunyasuvunakool *et al.*, “Highly accurate protein structure prediction for the human proteome,” *Nature*, vol. 596, no. 7873, pp. 590–596, Aug. 2021, doi: 10.1038/s41586-021-03828-1.
[28] M. C. Jespersen, B. Peters, M. Nielsen, and P. Marcatili, “BepiPred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes,” *Nucleic Acids Res*, vol. 45, no. Web Server issue, pp. W24–W29, Jul. 2017, doi: 10.1093/nar/gkx346.
[29] B. Petersen, T. N. Petersen, P. Andersen, M. Nielsen, and C. Lundegaard, “A generic method for assignment of reliability scores applied to solvent accessibility predictions,” *BMC Struct Biol*, vol. 9, no. 1, p. 51, Dec. 2009, doi: 10.1186/1472-6807-9-51.
[30] S. Altschul, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, Sep. 1997, doi: 10.1093/nar/25.17.3389.
[31] S. J. Hubbard and J. M. Thornton, “NACCESS.” Department of Biochemistry and Molecular Biology University College London, 1993.
[32] The NIH HMP Working Group *et al.*, “The NIH Human Microbiome Project,” *Genome Res.*, vol. 19, no. 12, pp. 2317–2323, Dec. 2009, doi: 10.1101/gr.096651.109.

Figure captions

Figure 1. Example of NetChop results which illustrate how to parse its results

Figure 2. TriTryp app screenshot of adding a new step in the search strategy, using the 1 MINUS 2 method.

Figure 3. Final search strategy, with individual nested strategies shown below.