Published in final edited form as:

Methods Mol Biol. 2020 ; 2120: 197–212. doi:10.1007/978-1-0716-0327-7_14.

Modeling and viewing T cell receptors using TCRmodel and TCR3d

Ragul Gowthaman^{1,2,3}, Brian G. Pierce^{1,2,3,#}

¹University of Maryland Institute for Bioscience and Biotechnology Research, Rockville, MD 20850, USA

²Department of Cell Biology and Molecular Genetics, University of Maryland, College Park, MD 20742, USA

³University of Maryland Marlene and Stewart Greenebaum Comprehensive Cancer Center, Baltimore, MD 21201, USA

Abstract

The past decade has seen a rapid increase in T cell receptor (TCR) sequences from single cell cloning and repertoire-scale high throughput sequencing studies. Many of these TCRs are of interest as potential therapeutics or for their implications in autoimmune disease or effective targeting of pathogens. As it is impractical to characterize the structure or targeting of the vast majority of these TCRs experimentally, advanced computational methods have been developed to predict their 3D structures and gain mechanistic insights into their antigen binding and specificity. Here we describe the use of a TCR modeling web server, TCRmodel, which generates models of TCRs from sequence, and TCR3d, which is a weekly-updated database of all known TCR structures. Additionally, we describe the use of RosettaTCR, which is a protocol implemented in the Rosetta framework that serves as the command-line backend to TCRmodel. We provide an example where these tools are used to analyze and model a therapeutically relevant TCR based on its amino acid sequence.

Keywords

TCR; Rosetta; MHC; antigen; bioinformatics; immunology

1. Introduction

T cell receptors (TCRs) are heterodimeric immunoglobulin proteins that play a key role in the immune system by mediating specific antigen recognition by T cells. They are composed of two different protein chains, most commonly α and β chains, while some are composed of γ and δ chains. TCRs engage foreign peptide or small molecule antigens bound to MHC (major histocompatibility complex) or MHC-like molecules, and productive binding events

[#]To whom correspondence should be addressed: pierce@umd.edu.

⁵The user must provide values for the two flags '-alpha' and '-beta'. The input sequences for α and β TCR chains need to be provided by the user separately as strings using the flags '-alpha' and '-beta'. All the other flags are optional flags. Other important flags used for running RosettaTCR are summarized in Table 4.

trigger T cell signaling, which cascades into various cellular responses from T cell development to apoptosis [1]. Due to their capability to specifically target tumor antigens, TCRs are increasingly being utilized as cell-based and soluble immunotherapeutics for cancer [2,3], with a number of candidates under investigation by the biopharmaceutical industry and progressing to the clinic [4]. TCRs recognize different classes of ligands presented by MHC or MHC-like molecules, including peptides presented by MHC class I or class II molecules, lipid antigens presented by CD1 and metabolites bound to MHC class I-related protein (MR1). Advances in high throughput sequencing techniques allow TCR sequences to be obtained in massive numbers, providing the opportunity to use advanced computational modeling tools to gain insights into their structure, recognition, and specificity. These can capitalize on the limited but steadily growing number of experimentally determined TCR structures available in the Protein Data Bank (PDB) [5], in conjunction with sequence-based feature recognition methods [6].

We recently developed a computational method to model TCR structures in atomic resolution based on their amino acid sequences, implemented as the TCRmodel web server [7]. This algorithm consists of three main stages: template identification, grafting and refinement, and is implemented in the Rosetta framework [8].

We also developed a database, TCR3d, which provides an interface to view features of experimentally determined TCR structures and their complexes [9]. The TCR3d site is divided into three main categories: TCR complexes, TCR structures and TCR sequences. Figure 1 shows the high-level organization of the database, including its core tables. Notable features of the database include the ability to search sequences and motifs of complementarity determining region (CDR) loops against known structures, graphical views of germline gene coverage in TCR structures, representations of all known docking geometries, and sequences of cancer-targeting and viral-targeting TCRs compiled from the literature.

In this chapter, we describe information and examples on the use of the TCRmodel web server, as well as the RosettaTCR protocol for modeling TCRs, which is the command-line implementation of TCRmodel. We also highlight the interface and use of the TCR3d database, including browsing features and query-based searches. The availability of these methods is given in Table 1.

2. Materials

2.1 Hardware/Software

RosettaTCR was developed as a protocol for modeling TCRs within the Rosetta modeling suite, which can be obtained for free by academic users and installed on Windows, Linux, or Mac OS. The RosettaTCR protocol requires Rosetta to be compiled with full C++11 support. We recommend the use of GCC version 4.9 or higher, or Clang-version 3.6 or higher, to compile the Rosetta source code. The application name is “tcrmodel”, which after compilation can be used as a command line program.

The TCR modeling application is available as a web server with an easy to use interface (<https://tcrmodel.ibbr.umd.edu/>), and any commonly used web browsers can be used to access it. This web server provides an interface to RosettaTCR, along with an up-to-date database of template framework and loop structures for modeling, and the capability to input sets of TCRs for batch processing.

TCR3d was developed with SQLite relational database management system. The web interface for the database is developed using the Python-based microframework, Flask. The database can be accessed at <https://tcr3d.ibbr.umd.edu/>, and is compatible with all commonly used web browsers. The entire database can also be downloaded and accessed locally using DB Browser for SQLite (<https://github.com/sqlitebrowser/sqlitebrowser>). All the structures present in the database and the complexes files can also be downloaded by the user (see Note 1).

2.2 Input template data

The TCR template database, which is required for running RosettaTCR, is not available by default for weekly Rosetta releases. The TCR template database directory is located at “Rosetta/database/additional_protocol_data/tcr” in the code base. Users can clone or download the template database as part of the “Rosetta/database/additional_protocol_data” directory.

For example:

```
git clone git@github.com:RosettaCommons/additional_protocol_data.git
```

or

```
git clone https://github.com/RosettaCommons/additional_protocol_data.git
```

The template database can also be downloaded from TCRmodel web server (see Note 2).

3. Methods

3.1 TCR3d database

3.1.1 TCR chains—The individual α , β , δ , and γ chains of publicly available TCR structures are provided as separate tables in TCR3d. In each table, the PDB code, chain ID, species, and the germline gene information for the TCR chains are available, as well as CDR1, CDR2 and CDR3 loop amino acid sequences.

3.1.2 Germline genes—The current human and mouse TCR variable gene coverage by TCR structures is shown in the ‘Germline Genes’ section. The list of PDB structures

¹The structures and complexes present in TCR3d are available for download. The structures of TCR chains, TCR complexes and the docking benchmark sets can be downloaded separately. The entire database in the SQLite3d format is also available for download. All the download links can be accessed from <https://tcr3d.ibbr.umd.edu/downloads>.

²The template database for the TCRmodel is updated regularly and is available for download from the TCRmodel web site from this page: <https://tcrmodel.ibbr.umd.edu/links>. It is also available in the Rosetta software suite.

available for each gene can be accessed by clicking the gene name on the x-axes of these plots.

3.1.3 TCR complexes—Complexes for all $\alpha\beta$ TCRs are classified according to antigen-presenting molecules TCRs, namely MHC Class I, MHC Class II, CD1, and MR1, while $\gamma\delta$ TCR complexes are shown separately. These tables include TCR name, MHC type, species, TCR germline gene and the epitope information.

3.1.4 Interface analysis—TCR complex geometries such as crossing angles and incident angles are calculated using a standalone program (Table 1) for each complex structure, while shape complementarity and interface size are calculated separately. These values can be obtained for each TCR complex from the corresponding tables, and are also plotted as figures for easy reference.

3.1.5 Structural viewing options—By clicking on the PDB code in the complexes table, users access the complex structure viewer showing the TCR, antigen, and MHC protein. Example structures for TCR-CD1d (PDB code 3ARB) and TCR-MHC Class II (PDB code 1D9K) complexes are shown in Figure 2.

3.1.6 CDR clusters—TCR CDR loop structures were clustered based on backbone conformational distances and affinity propagation, analogous to a previously described method for antibody CDR loop clustering [10]. These clusters can be accessed from TCR3d under the “Structures” heading. The CDR clusters section of the database provides the cluster table that list all the clusters identified with this method, showing CDR loop type, sequence length, number of structures in each cluster, and the representative sequence logo. Each cluster can be viewed separately in more detail to show the PDB IDs, sequences and the structural alignment of the CDR loops present in the cluster. Figure 3 shows the CDR clusters page and the cluster information and structural alignment for the cluster B_cdr3_11_C2 (CDR3 β cluster number 2 with sequence length 11).

3.1.7 TCR sequences—Curated sets of TCR sequences that target cancer and viral peptide antigens have been collected from the recent literature and are available in the database. Given their relevance in cancer immunotherapy, sequences of TCRs that target cancer-associated neoepitopes are included in the database, and neoepitope information is given if available. These tables contain links to model individual TCR sequences using the TCRmodel server.

3.1.8 TCR docking benchmark set—We have assembled sets of TCR complexes along with separately determined structures of unbound TCRs and peptide-MHCs, which represents an updated set of cases from a previously described predictive TCR docking benchmark [11]. There are currently 30 cases available in this set, and information on binding conformational changes is provided for each case.

3.1.9 CDR3 sequence search—The Sequence Search page allows users to search against TCR variable domain sequences, CDR3 sequences, and peptide antigens present in complex structures. Query CDR3 sequences are searched against all the CDR3 sequences

present the TCR structures and against the cancer/virus targeting TCR sequences. The input query is also searched for matches in the VDJdb database, which contains a large number of TCR sequences with known antigen specificity [12]. Sequence matches are scored and sorted by sequence similarity, using the PAM30 substitution matrix [13].

3.1.10 Subsequence or motif search—By default, only CDR3 sequences of the same length as the query are identified during searches. The user can optionally perform a subsequence search to identify specific sequence motifs present in CDR3 sequences. Figure 4 shows the sequence search page with results for a full length CDR3 sequence search (Input query from the TCRmodel example described below: CDR3 β sequence CASSQNYEQYF) and for a CDR3 subsequence search (Input query sequence: “GNQF”) representing a shared “public” CDR3 α motif found in TCRs that target a peptide epitope from human cytomegalovirus (HCMV) [14].

3.1.11 Peptide sequence search—The user can search for peptide sequence against all the peptides present in TCR complex structures. The exact or closest matches to the peptide present in structurally characterized TCR complexes are listed along with sequence alignments. As with the CDR3 sequence search, matching peptide sequences are scored and sorted using the PAM30 substitution matrix, and subsequence searches are provided as an option.

3.1.12 TCR full length sequence search—Users can search using a variable domain TCR sequence against all the TCR structures available in the database.

3.2 Modeling using the TCRmodel web server

TCRmodel takes α and β chain amino acid sequences as input. The input sequences must contain the variable domains of the TCR, including the CDR1, CDR2 and CDR3 loops. There are different ways that users can submit these input sequences.

In the following example, we use a recently described TCR sequence identified in a melanoma patient (“TCR 53”), that is specific to a neoantigen CDK4 peptide [15], to demonstrate the TCRmodel method. This TCR sequence is present in the Sequences section of the TCR3d database.

3.2.1 Input variable domain sequence—In TCRmodel, users can submit the variable domain sequences of α and β chains directly. The program is able to parse the variable domain sequence even if the user submits larger full-length variable and constant domain sequences.

Example α and β chain sequences:

>alpha

VTQLDSHVSVSEGTPVLLRCNYSSSYSPSLFWYVQHPNKGLQLLLKYTSAATLVKGI
NGFEAEFFKSETSFHLTKPSAHMSDAAEYFCVVSDLYNTDKLIFGTGTRLQVFP

>beta

VSQSPRYKVAKRGQDVALRCDPISGHVSLFWYQQALGQGPEFLTYFQNEAQLDKSG
LPSDRFFAERPEGSVSTLKIQRQQEDSAVYLCASSQNYEQYFGPGTRLTVT

3.2.2 Generate from germline genes—As many studies report TCRs as germline genes and CDR3 sequences, users can optionally input TCRs in this format. The user can choose the V/J genes along with input CDR3 sequence for the server to generate the full-length variable domain sequences. The following information is used to input the sequence for the same example TCR as above.

TRAV gene: TRAV8-2

TRAJ gene: TRAJ34

CDR3 α sequence: CVVSDLYNTDKLIF

TRBV gene: TRBV7-8

TRBJ gene: TRBJ2-7

CDR3 β sequence: CASSQNYEQYF

3.2.3 TCRmodel server results—Once the modeling is completed the server redirects to the results page. On the results page, all the template information are displayed, the modeled structure can be viewed and a download link is provided for the user to download the modeled file. Each modeling job is given a unique job ID. The job ID can also be used to retrieve the results later. The results are stored on the server for a month. TCRmodel server results for the example TCR sequence are shown in Figure 5.

3.3 Modeling using the RosettaTCR protocol

3.3.1 Input sequences—The RosettaTCR protocol is implemented as a command line application named “tcrmodel” in Rosetta, and compared with the TCRmodel server, it has various additional options for the user to choose and control different steps in the modeling protocol. For this application, the input sequences for alpha and beta chains need to be submitted separately. Users should include the flags ‘-alpha’ and ‘-beta’ to provide the sequences for α and β chains respectively (*see* Note 4).

Command: \$Rosetta/main/source/bin/tcrmodel.linuxgccrelease -alpha
<alpha_chain_sequence> -beta <beta_chain_sequence>

3.3.2 Parsing TCR segments—RosettaTCR has different methods to parse the input sequences into variable domain, framework, CDR1 loop, CDR2 loop and CDR3 loop segments. The program uses regular expressions to parse the input sequences by default. For cases in which the parsing by regular expression fails, the user has other options to define the TCR segments. The TCRmodel application supports the ANARCI program for parsing

⁴.For advanced users, and for those who may need to model a large number of TCR sequences, there is a batch process option available in the TCRmodel web server.

the input sequence [16]. Users should use the flag ‘-anarci_path’ to provide the path of the ANARCI program.

3.3.3 Assign TCR segments manually—The user can also manually assign the CDR segments. The user needs to provide the residue start and end positions for the CDR loops. The following flags are used for assigning CDR loop positions for the above example sequence.

-assign_cdr

-assign_cdr1a 23:33

-assign_cdr2a 47:74

-assign_cdr3a 91:100

-assign_cdr1b 24:33

-assign_cdr2b 47:75

-assign_cdr3b 92:100

For the above example TCR sequence the following parsed segments were obtained.

TCR α chain variable domain:

*VTQLDSHVS VSEGTPVLLRCNYSSSYSPSLFWYVQHPNKGLQLLLKYTSAATLVKGI
NGFEAEFKKSETSFHLTKPSAHMSDAAEYFCVVSDLYNTDKLIFGTGTRLQVF*

TCR β chain variable domain:

*VSQSPRYKVAKRGQDVALRCDPISGHVSLFWYQQALGQGPEFLTYFQNEAQLDKSG
LPSDRFFAERPEGSVSTLKIQRQQEDSAVYLCASSQNYEQYFGPGTRLTVT*

TCR α chain framework sequence:

*VTQLDSHVS VSEGTPVLLRCWYVQHPNKGLQLLLTKPSAHMSDAAEYFCFGTGTRL
QVF*

TCR β chain framework sequence:

*VSQSPRYKVAKRGQDVALRCWYQQALGQGPEFLTKIQRQQEDSAVYLCFGPGTRL
TVT*

CDR1 α : *NYSSSYSPSLF*

CDR1 β : *DPISGHVSLF*

CDR2 α : *LKYTSAATLVKGINGFEAEFKKSETSFH*

CDR2 β : *TYFQNEAQLDKSGLPSDRFFAERPEGSVST*

CDR3 α : *CVVSDLYNTDKLIF*

CDR3 β : *CASSQNYEQYF*

3.3.4 Template selection—Templates are selected for each segment from the template database, identifying matching segments of the same sequence length by PAM30 score. For modeling, TCR segments are grouped into one of the following sets:

Framework segment, CDR1, CDR2 and CDR3

Germline segment, CDR3

The program selects templates by default using best sequence match for germline or framework segments. If the program can find an exact template sequence for a given germline segment (including both CDR1 and CDR2 loops), then the germline template is assigned, and will be used to generate the modeled variable domain in conjunction with a CDR3 template. If no exact germline template sequence is found, then individual templates are assigned for framework segment, CDR1, CDR2 and CDR3. The user can force RosettaTCR to use full germline templates using the flags ‘-use_alpha_germline_templates’ and ‘-use_beta_germline_templates’ for alpha and beta chains respectively. The important flags used for template selection are summarized in Table 2.

3.3.5 User-provided templates—Users can specify templates structures to use for any segment. The following flags are used to providing templates structures for the α chain sequence:

-use_alpha_germline_templates <true>

-alpha_germline_template_pdb <germline_structure.pdb>

-alpha_cdr3_template_pdb <cdr3a_structure.pdb>

The user can choose particular template structures by providing the PDB four letter code with chain ID, and the corresponding PDB structure should be present in the template database. The following flags are used for providing template PDB names for the beta chain sequence (example PDB and chain IDs shown):

-use_beta_germline_templates <true>

-beta_germline_template_id 2Q86_B

-beta_cdr3_template_id 3PQY_E

3.3.6 Blacklist templates—The user can blacklist specific templates to prevent their use in modeling. The list of PDB IDs that should be blacklisted should be added to a file and provided to the program using the flag ‘-ignore_list’. Users can also remove templates based on percent sequence identity between template and target sequence, or ignore templates based on sequence identity. Corresponding flags are:

-ignore_list <blacklist.txt>

-template_identity_cutoff <90>

-blastp_identity_cutoff Identity cutoff <85>

3.3.7 Loop refinement—Loop modeling and refinement methods are integrated within the RosettaTCR protocol. The ‘remodel’ flag is used to reconstruct the loop segment by sampling the backbone conformations, whereas ‘refine’ flag is used to find the low energy conformations of the loop segment from the given starting conformation. The loop refinement can be applied to any particular CDR loop segment of the α or β chains, or only to CDR3 loops. The different options available for loop refinement are summarized in Table 3.

3.3.8 Interpreting the Output—1. Running RosettaTCR will result in the following PDB format output files, which are stored in the working directory. The -out::prefix flag can be used to add prefixes to the output file names. tcrmodel.pdb, the final modeled structure file.

tcr_graftmodel.pdb, the initial grafted model without refinement.

tcr_loopmodel.pdb, the model after optional loop modeling or refinement.

2. The various template structures used in the modeling can be obtained by using the flag ‘-dump_templates’.

3. The final modeled structure is scored by the Rosetta scoring function and the score is shown in the terminal output log. The score values are also included in the PDB file. The REF15 function [17] is the current scoring function used in Rosetta to score the structures. Along with the score, all the parsed TCR segment and the template information are displayed as output on the terminal.

3.4 Example application and results

Here we described an example input command and output log of tcrmodel application. A demo for the application is included in the Rosetta software suite (see Note 3). To run the TCRmodel application on command line the commands and the corresponding output log is given below:

Command:

```
$Rosetta/main/source/bin/tcrmodel.linuxgccrelease -alpha
VTQLDSHVSVSEGTPVLLRCNYSSSYSPSLFWYVQHPNKGLQLLLKYTSAATLVKGI
NGFEAEFKKSETSFHLTKPSAHMSDAAEYFCVSDLYNTDKLIFGTGTRLQVFP -beta
VSQSPRYKVAKRGQDVALRCDPISGHVSLFWYQQALGQGPEFLTYFQNEAQLDKSG
```

³A Rosetta demo and tutorial for RosettaTCR with an example is also available within the Rosetta tutorial section. The demo can be accessed from \$Rosetta/demos/public/tcr_modeling. This directory contains the example commands for running the tcrmodel application along with input and output files. The various optional flags used during modeling are explained and included in the demo directory.

LPSDRFFAERPEGSVSTLKIQRQQEDSAVYLCASSQNYEQYFGPGTRLTVT -
minimize_model true

The terminal output for the example sequence is shown below. Note that users' output details may differ due to possible differences in template library, etc.

termmodel score: -316.467

Tcr Alpha truncated Domain sequence:

VTQLDSHVSVSEGTPVLLRCNYSSSYSPSLFWYVQHPNKGLQLLLKYTSAATLVKGI
NGFEAEFKKSETSFHLTKPSAHMSDAAEYFCVSDLYNTDKLIFGTGTRLQVF

Tcr Alpha germline sequence :

VTQLDSHVSVSEGTPVLLRCNYSSSYSPSLFWYVQHPNKGLQLLLKYTSAATLVKGI
NGFEAEFKKSETSFHLTKPSAHMSDAAEYFCFGTGTRLQVF

Tcr Alpha Framework sequence :

VTQLDSHVSVSEGTPVLLRCWYVQHPNKGLQLLLTKPSAHMSDAAEYFCFGTGTRL
QVF

Tcr Alpha Domain CDR1 sequence: NYSSSYSPSLF

Tcr Alpha Domain CDR2 sequence: LKYTSAATLVKGINGFEAEFKKSETSFH

Tcr Alpha Domain CDR3 sequence: VVSDLYNTDKLI

Alpha Framework template : 6fr7_A_A

Alpha CDR1 template : 6fun_A_A

Alpha CDR2 template : 6fun_A_A

Alpha CDR3 template : 4jrx_D_A

Tcr Beta truncated Domain :

VSQSPRYKVAKRGQDVALRCDPISGHVSLFWYQQALGQGPEFLTYFQNEAQLDKSG
LPSDRFFAERPEGSVSTLKIQRQQEDSAVYLCASSQNYEQYFGPGTRLTVT

Tcr Beta germline sequence :

VSQSPRYKVAKRGQDVALRCDPISGHVSLFWYQQALGQGPEFLTYFQNEAQLDKSG
LPSDRFFAERPEGSVSTLKIQRQQEDSAVYLCFGPGTRLTVT

Tcr Beta Framework sequence:

VSQSPRYKVAKRGQDVALRCWYQQALGQGPEFLTKIQRQQEDSAVYLCFGPGTRL
TVT

Tcr Beta Domain CDR1 sequence: DPISGHVSLF

Tcr Beta Domain CDR2 sequence: TYFQNEAQLDKSGLPSDRFFAERPEGSVST

Tcr Beta Domain CDR3 sequence: ASSQNYEQY

Beta germline template : 1kgc_E_B

Beta CDR3 template : 3pqy_E_B

Alpha Beta orientation template : 4ww2_A_A 4ww2_B_B

Output grafted model: tcrgraftmodel.pdb

Output final model: tcmodel.pdb

References

1. Gaud G et al. (2018) Regulatory mechanisms in T cell receptor signalling. *Nat Rev Immunol* 18 (8):485–497. doi:10.1038/s41577-018-0020-8 [PubMed: 29789755]
2. Hinrichs CS, Rosenberg SA (2014) Exploiting the curative potential of adoptive T-cell therapy for cancer. *Immunol Rev* 257 (1):56–71. doi:10.1111/imr.12132 [PubMed: 24329789]
3. Liddy N et al. (2012) Monoclonal TCR-redirected tumor cell killing. *Nat Med* 18 (6):980–987. doi:10.1038/nm.2764 [PubMed: 22561687]
4. Zhang J, Wang L (2019) The Emerging World of TCR-T Cell Trials Against Cancer: A Systematic Review. *Technol Cancer Res Treat* 18:1533033819831068. doi:10.1177/1533033819831068
5. Rose PW et al. (2011) The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic acids research* 39 (Database issue):D392–401. doi:10.1093/nar/gkq1021 [PubMed: 21036868]
6. Glanville J et al. (2017) Identifying specificity groups in the T cell receptor repertoire. *Nature* 547 (7661):94–98. doi:10.1038/nature22976 [PubMed: 28636589]
7. Gowthaman R, Pierce BG (2018) TCRmodel: high resolution modeling of T cell receptors from sequence. *Nucleic Acids Res* 46 (W1):W396–W401. doi:10.1093/nar/gky432 [PubMed: 29790966]
8. Leaver-Fay A et al. (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in enzymology* 487:545–574. doi:10.1016/B978-0-12-381270-4.00019-6 [PubMed: 21187238]
9. Gowthaman R, Pierce BG (2019) TCR3d: The T cell receptor structural repertoire database. *Bioinformatics*, In Press doi:10.1093/bioinformatics/btz517
10. North B et al. (2011) A new clustering of antibody CDR loop conformations. *Journal of molecular biology* 406 (2):228–256. doi:10.1016/j.jmb.2010.10.030 [PubMed: 21035459]
11. Pierce BG, Weng Z (2013) A flexible docking approach for prediction of T cell receptor-peptide-MHC complexes. *Protein Sci* 22 (1):35–46. doi:10.1002/pro.2181 [PubMed: 23109003]
12. Shugay M et al. (2018) VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Res* 46 (D1):D419–D427. doi:10.1093/nar/gkx760 [PubMed: 28977646]
13. Dayhoff M et al. (1978) A model of evolutionary change in proteins In: *Atlas of protein sequence and structure*, vol 5 National Biomedical Research Foundation Silver Spring, pp 345–352
14. Yang X et al. (2015) Structural Basis for Clonal Diversity of the Public T Cell Response to a Dominant Human Cytomegalovirus Epitope. *J Biol Chem* 290 (48):29106–29119. doi:10.1074/jbc.M115.691311 [PubMed: 26429912]
15. Stronen E et al. (2016) Targeting of cancer neoantigens with donor-derived T cell receptor repertoires. *Science* 352 (6291):1337–1341. doi:10.1126/science.aaf2288 [PubMed: 27198675]
16. Dunbar J, Deane CM (2016) ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics* 32 (2):298–300. doi:10.1093/bioinformatics/btv552 [PubMed: 26424857]
17. Alford RF et al. (2017) The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J Chem Theory Comput* 13 (6):3031–3048. doi:10.1021/acs.jctc.7b00125 [PubMed: 28430426]

Figure 1.

The TCR3d database structure and contents. The database includes curated sets of structures and sequences (“Datasets”, upper boxes) as well as derived data and search features (“Analysis”, lower boxes), and is organized into three major categories: 1. TCR complexes, which include TCRs in complex with various MHC and MHC-like molecules and antigens, calculated properties of complex structures, and a docking benchmark set. 2. TCR structures, which includes details on all TCR chains in the PDB, overall structural coverage of TCR germline genes, and CDR loop clusters. 3. TCR sequences, which include virus-targeting and cancer-targeting TCR sequences from the literature, along with options to perform sequence searches against TCR structures and sequences.

Figure 2.

TCR3d views of a complex structures for a (A) TCR-glycolipid-CD1 complex (PDB: 3HUJ) and (B) a TCR-peptide-Class II MHC complex (PDB: 3O6F). TCR chains are blue and green, CD1 is red, Class II MHC is red and yellow, and antigenic glycolipid and peptide are shown as gray sticks. TCR orientation axes and MHC or MHC-like antigen-binding plane are magenta.

Figure 3. Viewing CDR clusters in TCR3d. (A) Table of all CDR clusters, and (B) cluster details and structural alignment for the cluster A_cdr3_10_C7.

Figure 4. Sequence search page in TCR3d. Shown are (A) Query interface, (B) results from CDR3 full length sequence search (query sequence: “CASSQNYEQYF”), and (C) results from CDR3 motif/subsequence search (query sequence: “GNQF”).

Figure 5. TCRmodel interface and results. (A) Input sequences entered in the TCRmodel web server, (B) identified templates and alignment, and (C) final modeled structure.

Table 1.

TCR tools and availability.

Name	Description	Web site
TCRmodel web server	Web server to model TCR structures from sequence	https://tcrmodel.ibbr.umd.edu/
RosettaTCR	Protocol in the Rosetta modeling suite to model TCR structures from sequence	https://www.rosettacommons.org
TCR3d	T cell receptor structural repertoire database	https://tcr3d.ibbr.umd.edu/
TCR docking angle program	Program to calculate TCR docking angle and incident angles	https://github.com/piercelab/tcr_docking_angle

Table 2.

Important flags used for template selection

Flag(s)	Description
-template_identity_cutoff	Identity cutoff to ignore template sequences from template database
-blastp_identity_cutoff	Cutoff to ignore similar template sequences from template database based on blast search.
-alpha_cdr1_template_id -alpha_cdr2_template_id -alpha_cdr3_template_id -alpha_framework_template_id -alpha_germline_template_id -alpha_orientation_template_id	Template PDB id with chain for alpha chain CDR1, CDR2, CDR3, Framework, germline segment and for orientation respectively.
-beta_cdr1_template_id -beta_cdr2_template_id -beta_cdr3_template_id -beta_framework_template_id -beta_germline_template_id -beta_orientation_template_id	Template PDB id with chain for beta chain CDR1, CDR2, CDR3, Framework, germline segment and for orientation respectively.
-alpha_cdr1_template_pdb -alpha_cdr2_template_pdb -alpha_cdr3_template_pdb -alpha_framework_template_pdb -alpha_germline_template_pdb -alpha_orientation_template_pdb	Template PDB structure file for alpha chain CDR1, CDR2, CDR3, Framework, germline segment and for orientation respectively.
-beta_cdr1_template_pdb -beta_cdr2_template_pdb -beta_cdr3_template_pdb -beta_framework_template_pdb -beta_germline_template_pdb -beta_orientation_template_pdb	Template PDB structure file for beta chain CDR1, CDR2, CDR3, Framework, germline segment and for orientation respectively.

Table 3.

Important TCRmodel flags used for loop refinement

Flag	Description
-remodel_tcr_cdr3a_loop	remodel the cdr3 loop of alpha chain
-remodel_tcr_cdr3b_loop	remodel the cdr3 loop of beta chain
-refine_tcr_cdr3a_loop	refine the cdr3 loop of alpha chain
-refine_tcr_cdr3b_loop	refine the cdr3 loop of beta chain
-remodel_tcr_cdr3_loops	remodel the cdr3 loop of alpha and beta chains
-refine_tcr_cdr3_loops	refine the cdr3 loop of alpha and beta chains
-refine_all_tcr_cdr_loops	refine the cdr1, cdr2 and cdr3 loops of alpha and beta chains

Table 4.

Summary of important TCRmodel flags used for modeling.

Flag	Default	Description
-alpha	None	Amino acid sequence of the Sequence of TCR alpha chain.
-beta	None	Amino acid sequence of the Sequence of TCR beta chain.
-minimize_model	True	Minimize the output model.Default:True
-relax_model	False	Relax the output model.
-ignore_list	None	List of PDB id's to ignore as templates