The detection of T-cell epitopes is a critical step in vaccine design. Tools for epitope detection are available today do not offer tools for flexible data processing. Since prediction methods becomes increasingly important. These pipelines extendable by user-defined prediction methods or methods for filtering of results. FRED is implemented in Python (release 2.6) (www.python.org). Additional software required for FRED is freely available and installation packages are included in the FRED package. The prediction methods currently available in FRED are listed in Table 1. 

2 METHODS

2.1 Implementation

FRED provides methods for sequence input, sequence preprocessing, filtering and display of the results. The general organization of FRED is depicted in Figure 1. The single prediction methods are accessed internally via a consistent interface. FRED can handle polymorphic sequences (e.g. for the study of single nucleotide polymorphisms (SNPs) in an epitope context) and offers the possibility of accessing different methods simultaneously and of combining, comparing or benchmarking the methods. FRED is easily extendable by user-defined prediction methods or methods for filtering of results.

3 APPLICATIONS

Tutorial and documentation: with the FRED package, we provide examples that demonstrate how FRED can be used to solve typical tasks in computational immunomics with short and simple scripts. A detailed tutorial is available on the project’s web site. It explains how to implement prediction pipelines, offers more detailed information on the functionality of FRED and addresses problems like choosing the right prediction method or threshold. We additionally provide a detailed documentation of the code.

Vaccine design: the selection of peptides for epitope-based vaccines is a typical application for large-scale predictions of MHC binding peptides. The following short and simple program implements a short example of FRED's capabilities:

```python
# Selection of peptides for epitope-based vaccines

from FRED import FRED

# Define prediction methods
methods = ["Method1", "Method2", "Method3"]

# Load data
peptide_sequences = load_peptide_data("sequence_file.txt")

# Use FRED to predict epitopes
epitopes = FRED.predict(epitopes, peptide_sequences, methods)

# Display results
display_results(epitopes)
```

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Table 1. Prediction methods currently integrated in FRED

| Method                   | References                                      |
|--------------------------|-------------------------------------------------|
| MHC binding:             |                                                 |
| SYFPEITHI                | Rammensee et al. (1999)                         |
| SVMHC                    | Dönnes and Kohlbacher (2006)                   |
| BIMAS                    | Parker et al. (1994)                           |
| NetMHCpan                 | Nielsen et al. (2007)                          |
| NetMHC                   | Buus et al. (2003)                             |
| Hammer                   | Sturmilo et al. (1999)                         |
| NetMHCIIpan               | Nielsen et al. (2008)                          |
| Proteasomal Cleavage:    |                                                 |
| PCM method from WAPP     | Dönnes and Kohlbacher (2005)                   |
| TAP Transport:           |                                                 |
| SVMTP                    | Dönnes and Kohlbacher (2005)                   |
| Additive matrix method   | Doytchinova et al. (2004)                      |

Integration of new methods and performance evaluation: epitope prediction is still a very active field, with new methods continuously being developed. Such methods not implemented in Python can be plugged in using command-line calls. FRED provides a number of standard measures to compare different prediction methods and to evaluate the performance w.r.t. experimental values (Matthews Correlation Coefficient, accuracy, sensitivity, specificity, area under the ROC curve, correlation and rank correlation). Different prediction methods can thus be compared with ease.

Web server development: using FRED as the basis for new applications in computational immunomics leads to a significant reduction of development time and allows the convenient combination of new methods with existing ones. An example of an application based on FRED is EpiTookit (www.epitokit.org, Feldhahn et al., 2008; Toussaint and Kohlbacher, 2009). Only the web-based user interface and the data management in the web server had to be newly implemented. Prediction functionality of EpiTookit is completely provided by FRED. Through the use of Python, FRED can be integrated seamlessly in web servers/content management systems like Plone (http://www.plone.org/).

4 CONCLUSIONS

FRED is a valuable tool for performing large-scale analyses in immunoinformatics with different prediction methods and is also a software framework for the development of novel immunoinformatics methods. Ease of use, extendability and openness make it an ideal tool for addressing complex immunoinformatics problems in an uncomplicated manner.

References: Deutsche Forschungsgemeinschaft (SFB 685/B1).
Conflict of Interest: none declared.

REFERENCES

Buus, S. et al. (2003) Sensitive quantitative predictions of peptide-MHC binding by a “Query by Committee” artificial neural network approach. *Tissue Antigens*, 62, 378–384.
DelLucia, D. S. and Blasczyk, R. (2007) The immunoinformatics of cancer immunotherapy. *Tissue Antigens*, 70, 265–271.
Dönnes, P. and Kohlbacher, O. (2005) Integrated modeling of the major events in the MHC class I antigen processing pathway. *Protein Sci.*, 14, 2132–2140.
Dönnes, P. and Kohlbacher, O. (2006) SVMHC: a server for prediction of MHC-binding peptides. *Nucleic Acids Res.*, 34, W194–W197.
Doytchinova, I. et al. (2004) Transporter associated with antigen processing preselection of peptide binding to the MHC: a bioinformatic evaluation. *J. Immunol.*, 173, 6813–6819.
Feldhahn, M. et al. (2008) EpiTookit—a web server for computational immunomics. *Nucleic Acids Res.*, 36, W516–W522.
Halling-Brown, M. et al. (2009) Computational grid framework for immunological applications. *Philos Trans. R. Soc. A*, 367, 2705–2716.
Nielsen, M. et al. (2007) NetMHCpan: a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. *PLoS ONE*, 2, e796.
Nielsen, M. et al. (2008) Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan. *PLoS Comput. Biol.*, 4, e1000107.
Parker, K.C. et al. (1994) Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *J. Immunol.*, 152, 163–175.
Rammensee, H. et al. (1999) SYFPEITHI: database for MHC ligands and peptide motifs. *Immunogenetics*, 50, 213–219.
Sturmilo, T. et al. (1999) Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. *Nat. Biotechnol.*, 17, 555–561.
Toussaint, N.C. and Kohlbacher, O. (2009) OptiTope—a web server for the selection of an optimal set of peptides for epitope-based vaccines. *Nucleic Acids Res.*, 37, W617–W622.
Toussaint, N.C. et al. (2008) A mathematical framework for the selection of an optimal set of peptides for epitope-based vaccines. *PLoS Comput. Biol.*, 4, e1000246.