SVM-basierte Methode zur Vorhersage möglicher MHC-I-Epitope

Rehan App, Julian Späth & Kevin Krumm

Universität Tübingen
app.rehan@yahoo.de , spaethju@posteo.de, k.krumm@arcor.de

Abstract

MHC-Klasse-I Moleküle spielen seit einigen Jahren eine wichtige Rolle in der Entwicklung von Impfstoffen. Mit ihnen ist es möglich, die derzeit immer häufiger verwendeten Epitop-basierten Impfstoffe einerseits sehr effizient zu machen und andererseits die Nebenwirkungen von klassischen Impfstoffen zu umgehen. Mit Hilfe von maschinellen Lernverfahren, hier einer Support Vektor Maschine (SVM), lassen sich effiziente und genaue Vorhersagen für mögliche MHC-I-Epitope machen. Die hier vorgestellte Methode sagt für 9-mer Peptide voraus, ob sie an einem MHC-I-Molekül binden oder nicht. Das Programm wurde in Python mit scikit-learn umgesetzt und generiert aus einer Input-Datei mit Peptiden der Länge 9 eine Output-Datei, in der jedem der eingegebenen Peptiden die Eigenschaft "Binder" (1) oder "Nicht-Binder" (0) zugeordnet wird. Die SVM wurde auf einem Trainingsdatensatz für ein unbekanntes HLA-Klasse-1 Allel mit 727 9-mer Peptiden trainiert und hat mit einem mittleren AUC-Wert von 0.85 eine relativ hohe Vorhersagegenauigkeit.

I. Introduction

Impfstoffe sind schon seit einiger Zeit ein wichtiger Bestandteil im Kampf der Menschen gegen Krankheiten. Seit Edward Jenner 1788 die erste Pocken-Impfung entdeckte, wurden etliche neue Impfstoffe hergestellt, die Menschen vor Krankheiten beschützen. Das immer breitere Wissen über das Immunsystem fließt vehement in die Impfstoffentwicklung mit ein. So wurde unter anderem der MHC-Klasse-I-Antigenprozessierungspfad entschlüsselt, welcher intrazelluläre Antigene an der Zelloberfläche und somit den Zellen des Immunsystem präsentiert: Peptide werden im Zellinneren durch die Protease in kleine Fragmente gespaltet, von denen einige durch den TAP-Transporter in das endoplasmatische Retikulum gelangen. Dort wartet ein MHC-Molekül (Major Histocompatibility Complex) auf ein spezifisches Peptid-Fragment der Länge 8-10, welches es dann an der Zelloberfläche den Tc-Zellen präsentieren kann. Dadurch erkennt das Immunsystem, was in der Zelle vor sich geht und ob es dort eingreifen muss [1].

Lange Zeit wurden Impfstoffe hauptsächlich aus getöteten oder inaktiven, kompletten Organismen hergestellt. Das Ziel einer Immunisierung kann allerdings auch durch weitaus kleineren Fragmenten des Organismus erreicht werden, da es verschiedene Möglichkeiten gibt, dem Immunsystem ein Epitop zu präsentieren. So gewannen Epitop-basierte Vaccine aus MHC-bindenden Peptiden in den letzten Jahren immer mehr an Bedeutung. Je weniger eines Organismus dem Körper injiziert werden muss, desto geringer ist das Risiko für eine Infizierung mit der Krankheit, gegen die eigentlich immunisiert werden soll. Es reicht also aus kleinere Peptide zu impfen, welche dann an MHC-I binden. Die Identifizierung von MHC-I-bindenden Peptiden für die Impfstoffentwicklung ist somit auch eine große Herausforderung für die Immunoinformatik geworden. [2]

Für die Vorhersage dieser Peptide haben sich Maschinelle Lernverfahren, wie Neuronale Netze oder auch Support Vektor Maschinen, als sehr gute Methoden herauskristallisiert. Diese Verfahren lernen von einem Datensatz und stellen dadurch eine Verbindung zwischen den Klassen her. Aufgrund dieser Erfahrung

können diese Klassifikatoren dann neue Daten, die sie noch nie zuvor gesehen haben, wieder klassifizieren Die in den folgenden Abschnitten vorgestellte Methode zur MHC-I-Epitop Vorhersage konzentriert sich auf 9-mer Peptide. Als Klassifikator wurde eine SVM verwendet.

II. Materials & Methods

Table 1: *Programme & Frameworks*

Name	Version
Python	2.7.11
Numpy	1.11.0
SciPy	0.16.1
Scikit-learn	0.17.0
Matplotlib	1.5.1

Data

Als Trainingsdatensatz für die Methode diente ein schon vorgegebener Datensatz für ein unbekanntes HLA-Klasse-1 Allel. Dieser stellt 727 Peptide der Länge 9 sowie deren IC50-Wert und ihre Klassifikation dar. Ein Peptid bindet an MHC-I, wenn es mit einer 1 gelabelt ist. Eine 0 hingegen bezeichnet das Peptid als Nicht-Binder. Der IC50-Wert ist für das Training unserer Methode ohne Bedeutung. Der Datensatz ist mit 176 Binder und 551 Nicht-Binder sehr unausgeglichen, wodurch einige Anpassungen nötig wurden.

Zu Beginn wurden die Trainingsdaten in das richtige Format gebracht. Hierfür wurde jede Aminosäure für jedes Peptid des Trainingsdatensatzes als Feature-Vektor dargestellt. Diese Form der Daten wird von einer SVM, aber auch von vielen anderen maschinellen Lernverfahren erwartet. Als Features boten sich die Aminosäureeigenschaften der *amino acid index database* [3] an. Die Datenbank enthält derzeit 544 Indizes von Aminosäuren und ordnet jeder Aminosäure für jeden Index, zum Beispiel dem Hydrophobizitätsindex, einen Wert zu.

Um bessere und stabilere Ergebnisse zu erhalten, wurden die aus den daraus erstellten Feature-Vektoren noch mit einem Min-Max-Skalierer in den Bereich [0,1] skaliert. Dieser lieferte deutlich bessere Ergebnisse als der Max-Abs-Skalierer, welcher die Daten in einen Bereich von [-1, 1] skaliert hätte. Zusätzlich wurden die Features noch durch eine univariate Feature-Selektion gefiltert. Hierfür bot sich der Select Percentile Ansatz an. Dieser entfernte alle Features, bis auf diejenigen mit dem höchsten Scoring-Anteil. Dadurch wurden die irrelevanten Features herausgefiltert, so dass die Feature-Vektoren nur noch aus relevanten Features bestanden.

Klassifikator

Als Klassifikator für die MHC-I-Epitop-Vorhersage wurde eine Support Vektor Maschine (SVM) verwendet. Sie benötigt mehrere Parameter, welche optimiert werden müssen, um am Ende relevante und gute Ergebnisse zu erhalten.

Die in dieser Methode verwendete SVM benutzt einen radial-basis-function Kernel (rbf-Kernel). Da die MHC-I-Epitop-Vorhersage kein lineares Problem ist, würde ein linearer Kernel keine guten Ergebnisse liefern. Die Kernel-Funktion des rbf-Kernel lautet:

$$(-\lambda|x - x'|^2) \cdot \lambda$$

[4]. Die *Slackvariable* ξ erlaubt es, einzelne Objekte falsch zu klassifizieren. C bestraft eine solche Falschklassifizierung. Ein großes C bedeutet dabei weniger Toleranz für falsch klassifizierte Datenpunkte. Ein kleines C hingegen mehr Toleranz, da falsch klassifizierte Datenpunkte weniger bestraft werden. Der Parameter γ gibt an, wie weit der Einfluss eines einzelnen Trainingsbeispiels reicht [5]. Für eine optimale Wahl der Parameter für die SVM wurde eine *Grid-Search* durchgeführt. Außerdem wurde ein balanciertes Klassengewicht eingestellt, wodurch die Gewichte für den hier verwendeten, sehr unbalancierten Datensatz besser angepasst werden.

Training

Um die Genauigkeit des Klassifikators zu schätzen, wurde eine stratifizierte Kreuzvalidierung durchgeführt. Dadurch kann effektiv *overfitting* vermieden werden. Die stratifizierte Kreuzvalidierung hat einen bedeutenden Vorteil gegenüber der üblich gebräuchlichen K-Fold Kreuzvalidierung: Sie achtet darauf, dass jede der k Teilmengen eine Verteilung wie das komplette Set besitzt [6]. Dies erwies sich für den hier verwendeten, sehr unausgeglichenen Trainingsdatensatz als sehr nützlich. Schließlich teilte unsere Kreuzvalidierung den Datensatz in 10 kleinere Datensätze auf und für jeden konnte eine SVM mit Grid-Search durchgeführt werden. Dieser wurden einige plausible Werte für C und γ gegeben, woraufhin die Grid-Search schließlich die besten Werte herausfand und diese als Parameter in die SVM einsetzte. Für die Grid-Search wurde eine weitere 10-fache Kreuzvalidierung durchgeführt.

Die Bewertung der Qualität unserer Ergebnisse wurde durch die *ROC-Kurve* verwendet. Diese bewertet die Vorhersage anhand der *Area Under the Curve (AUC)*. Die ROC-Kurve plottet die True Positive Rate gegen die False Positive Rate und bietet somit einen Vergleich der Beiden. Die *AUC* ermöglicht eine etwas aussagekräftigere Qualitätsmessung bei unausgeglichenen Datensätzen. Sie beschreibt die Fläche unter der Kurve: Je weiter die Kurve nach links oben rückt, desto besser ist die Qualität der Vorhersage. Eine AUC von 0.5 kann jedoch auch durch Zufall erreicht werden und stellt somit keine zuverlässige Vorhersage dar. Ab etwa einer AUC von 0.8 kann man von einer recht guten Vorhersage sprechen [7].

Programm

Nachdem die ausführliche Kreuzvalidierung mit Grid-Search gute Ergebnisse geliefert hatte, wurde eine Grid-Search für die SVM auf dem kompletten Trainingsdatensatz durchgeführt. Die SVM mit den jeweils besten Werten für C und γ wurde trainiert und gespeichert und konnte anschließend für die Vorhersage von Peptiden der Länge 9 verwendet werden. Diese SVM wurde schließlich in ein Python-Konsolenprogramm verpackt, so dass die SVM leicht eine Datei mit 9-mer Liganden entgegennehmen kann. Als Output entsteht nach Ausführung des Programms eine Liste, mit den Liganden und dem vorhergesagten Label (1/0).

III. Results

Kreuzvalidierung

Nach ausführlicher Vorverarbeitung der Daten wurden durch die Stratified 10-Fold Kreuzvalidierung gute Ergebnisse erzielt.

Table 2: *Ergebnisse der Kreuzvalidierung*

Dataset	1	2	3	4	5
AUC	0.87	0.90	0.89	0.75	0.87
Dataset	6	7	8	9	10
AUC	0.89	0.76	0.86	0.83	0.87

In den 10 Folds, in denen jeweils eine Grid-Search ausgeführt wurde, erreichte die SVM im schlechtesten Fall eine AUC von 0.75 und im besten Fall sogar 0.9. Mit einem Mittelwert von 0.85 und einer ordentlichen mittleren ROC-Kurve der Durchgänge wurde ein gutes Ergebnis erzielt. Mit einer normalen 10-Fold Kreuzvalidierung hingegen waren die Ergebnisse etwas schlechter.

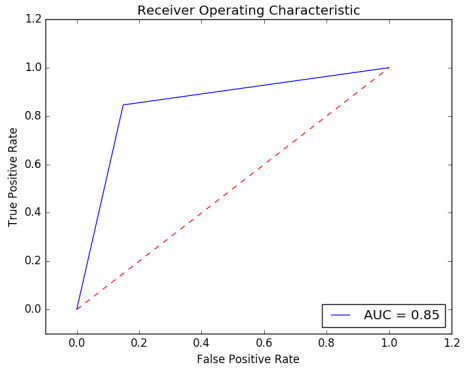

Figure 1: *ROC-Kurve der Kreuzvalidierung*

Aus der Kreuzvalidierung und der aus ihr entstandenen mittleren ROC-Kurve lässt sich schließen, dass die SVM auch auf anderen Datensätzen als auf dem Trainingsdatensatz gute Ergebnisse liefert. Es ist zu erwarten, dass die AUC des finalen Prädiktors im schlechtesten Fall nicht unter 0.75 liegt und sich eher im Bereich der mittleren AUC, also 0.85, befindet.

Finaler Prädiktor

Nach der Abschätzung der Genauigkeit durch eine Kreuzvalidierung mit Grid-Search konnte nun der finale Prädiktor erstellt werden, der später die Labels der Input-Datei vorhersagt. Dafür wurde eine Grid-Search auf dem kompletten Trainingsdatensatz durchgeführt. Die SVM mit den besten Parametern erreichte hier eine AUC von 0.9, also einen etwas höheren Wert als der, der in der Kreuzvalidierung erreicht wurde.

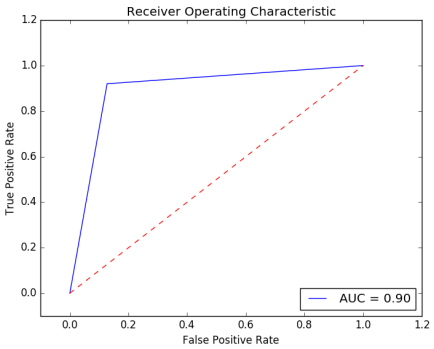

Figure 2: *ROC-Kurve der Kreuzvalidierung*

Diese SVM wurde trainiert und schließlich gespeichert, so dass sie im Programm leicht geladen werden konnte. Damit war eine Klassifikation des Inputs in Binder und Nicht-Binder durch die SVM möglich, ohne dass jedes mal eine neue SVM gebildet werden musste.

IV. Discussion

Die finale Version unseres Projekts stellt einen funktionsfähigen Prädiktor dar, welcher MHC-I-bindende Epitope vorhersagt. Peptide die hier als Binder gelabelt werden, eignen sich also möglicherweise als Epitop-Basierter Impfstoff. Es muss allerdings darauf geachtet werden, dass dieser Prädiktor nicht alle MHC-I-bindenden Epitope vorhersagen kann. An das MHC-I Molekül binden zwar vorwiegend, jedoch nicht ausschließlich 9-Mere. Dies bedeutet, dass mit diesem Programm mögliche Binder einer anderen Länge nicht vorhergesagt werden können. Außerdem wurde die Vorhersage für ein unbekanntes HLA-Klasse-I Molekül durchgeführt. Für den Gebrauch als wirkungsvollen Impfstoff müsste somit noch in Erfahrung gebracht werden, auf welchem HLA-Molekül die Daten wirklich beruhen, da in jeder Region unterschiedliche Wirksamkeiten auftreten.

Die ROC-Kurve und die Area Under the Curve zeigen, dass die Aussagekraft des Prädiktors hoch ist. Die AUC liegt mit 0.9 wesentlich über dem Zufallswert 0.5 und die ROC-Kurve somit weit über der ersten Winkelhalbierenden. Die Wahrscheinlichkeit für eine richtige Klassifizierung in Binder bzw. Nicht-Binder ist also sehr hoch. Die durch Feature-Selection gewählten Features, als auch die von der Grid-Search gefundenen Parameter der SVM scheinen gut gewählt zu sein. Die AUC des finalen Prädiktors ist allerdings mit einem Wert von 0.85 etwas höher als der mittlere AUC-Wert, der durch Kreuzvalidierung ermittelt wurde. Das liegt daran, dass auf dem eigenen Trainingsdatensatz meist bessere Ergebnisse erzielt werden als auf einem unabhängigen Datensatz. Für die Vorhersage des Inputs sind somit eher die Werte der Kreuzvalidierung zu beachten.

Wie man sieht, können maschinelle Lernverfahren zur Vorhersage von MHC-I-Epitopen verwendet werden und liefern auch gute Ergebnisse. Die hier vorgestellte SVM-basierte Methode kommt damit in der Genauigkeit der Vorhersage schon recht gut an schon vorhandene Prädiktoren, wie etwa SVMHC, heran. Mit diesen Verfahren wird nie eine Genauigkeit von 100% erreicht werden können, jedoch kann es die Impfstoffentwicklung und die damit verbundene Suche nach geeigneten MHC-Epitopen sehr gut unterstützen. Ein noch breit-

eres Wissen über die komplexen Vorgänge des Immunsystems sowie größere Datensätze mit mehr bekannten Bindern und Nicht-Bindern in Verknüpfung mit Maschinellen Lernverfahren könnten also in Zukunft noch wichtiger werden, als es derzeit schon ist, wenn die Menschheit Krankheiten wie Krebs besiegen will.

References

[1] Murphy, Kenneth. *Janeway's Immunobiology*. Garland Science, 2012, S.206-208

[2] Kohlbacher, Oliver. *Computational Immunomics - Vaccine Design*. Universität Tübingen, 2016, S.3-5

[3] Kyoto University Bioinformatics Center. *AAIndex - Amino acid indices substitution matrices and pair-wise contact potentials*. `http://www.genome.jp/aaindex/`, Stand: 23.06.2016

[4] Scikit-learn - Machine Learning in Python. *Support Vector Machines - Kernel Functions*. `http://scikit-learn.org/stable/modules/svm.html`, Stand: 25.06.2016

[5] Scikit-learn - Machine Learning in Python. RBF SVM parameters. `http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html`, Stand 25.06.2016

[6] Scikit-learn - Machine Learning in Python. Cross-validation: evaluating estimator performance - Cross validation iterators - Stratified k-fold. `http://scikit-learn.org/stable/modules/cross_validation.html`, Stand 25.06.2016

[7] Tape, Thomas G. - University of Nebraska Medical Center. *ROC-Curves - The Area Under an ROC Curve*. `http://gim.unmc.edu/dxtests/roc3.htm`, Stand 25.06.2016

[8] Hastie, Trevor & Tibshirani, Robert & Friedman, Jerome. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. Springer, 2008, S.417-423