Similarity Measure for Cell Membrane Fusion Proteins Identification

Daniela Megrian[1], Pablo S. Aguilar[2], and Federico Lecumberry[3](✉)

[1] Unidad de Bioinformática, Institut Pasteur de Montevideo, Montevideo, Uruguay
dmegrian@pasteur.edu.uy

[2] Laboratorio de Biología Celular de Membranas, IIBINTECH, CONICET,
Universidad Nacional de San Martín, San Martín, Argentina

[3] Departamento de Procesamiento de Señales, Instituto de Ingeniería Eléctrica,
Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
fefo@fing.edu.uy

Abstract. This work proposes a similarity measure between secondary structures of proteins capable of fusing cell membranes and its implementation in a classification system. For the evaluation of the metric we used secondary structures estimated from amino acid sequences of Class I and Class II viral fusogens (VFs), as well as VFs precursor proteins. We evaluated three different classifiers based on k-Nearest Neighbors, Support Vector Machines and One-Class Support Vector Machines in different configurations. This is a first approach to the similarity measure with satisfactory results. It is possible that this method could allow the identification of unknown membrane fusion proteins in other biological models than the proposed in this work.

Keywords: Cell membrane fusion · Viral fusogen · Similarity measure · Support Vector Machines · One-Class Support Vector Machines · k-Nearest Neighbors

1 Introduction

Fusion between cells is needed in many cellular events. Some of the most studied events are myoblasts fusion during muscle formation [1], fusion of gametes during fertilization [2] and fusion between extracellular vesicles (EVs) and target cells [3]. Cellular membranes cannot fuse spontaneously, this process is catalyzed by proteins named fusogens [4]. However, it is still unknown which proteins carry out the fusion mechanism during these events.

One of the best understood fusion mechanisms is the fusion between the membrane of an enveloped virus and the membrane of the target cell. The viral fusion proteins, or viral fusogens, can be grouped in at least three classes according to their structure and mechanism of action. Most of the known viral fusogens belong to Class I and II, that is why these classes are the better characterized. At secondary structure level, Class I viral fusogens present mostly α-helix structure, while Class II are organized mainly in β-sheet [5]. One of the few known

C. Beltrán-Castañón et al. (Eds.): CIARP 2016, LNCS 10125, pp. 257–265, 2017.
DOI: 10.1007/978-3-319-52277-7_32

cellular membrane fusion proteins is EFF protein from C. elegans. This protein is structurally homologous to Class II viral fusogens, also preserving the β-sheet secondary structure organization. In spite of this homology, the amino acid sequence highly differs from Class II viral fusogens sequences [6].

In this work, we intend to develop a similarity measure able to discriminate proteins with fusion capacity in different biological models, based on the proteins secondary structure.

In Sect. 2 we describe the previous attempts to find similarity between secondary structure sequences and its applications, including pattern recognition methods. In Sect. 3 the secondary structure alignment algorithm is explained, as well as the advantages of implementing it to our problem. The description of the data available and the experimental results are in Sect. 4. Section 5 concludes and presents some possible directions of work.

2 Background

The search for protein secondary structures alignment gathered strength with the appearance of reliable tools for secondary structure prediction from amino acid sequences such as described by Cuff et al. [7] and Mc Guffin et al. [8]. These tools return, for each amino acid position, an H (α-helix), E (β-sheet), or C (random coil) character corresponding to the most probable structure in that position, considering the propensities of individual amino acids (Fig. 1).

Most of the literature relative to secondary structure alignment is based on the method proposed by Przytycka et al. [9] called SSEA (Secondary Structure Element Alignment). In this method, the secondary structure of each protein is represented as a summarized and ordered sequence of characters H, E and C

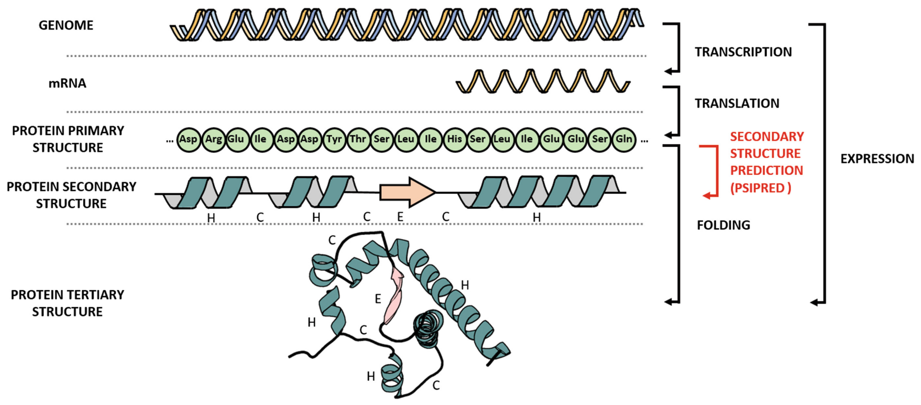
GENOME
mRNA
PROTEIN PRIMARY STRUCTURE
... Asp Arg Glu Ile Asp Asp Tyr Thr Ser Leu Ile His Ser Leu Ile Glu Glu Ser Gln ...
PROTEIN SECONDARY STRUCTURE
H C H C E C H
PROTEIN TERTIARY STRUCTURE
C
H
E
H
C
H
C
TRANSCRIPTION
TRANSLATION
SECONDARY STRUCTURE PREDICTION (PSIPRED)
FOLDING
EXPRESSION

Fig. 1. Context of secondary structure prediction. Gene expression is the process by which information contained in a genome is used to direct protein synthesis. The protein folds into a functional tridimensional molecule at two levels: secondary and tertiary structure.

(Fig. 2). The consecutive repeated characters are collapsed in an element, and the length of the element is stored. SSEA algorithm is analogous to the global alignment algorithm based on dynamic programming proposed by Needleman and Wunsch [10], but using a different score assignment system. When aligning two secondary structures X and Y, a score is calculated for each pair of elements x and y. Each score $S(x, y)$ is defined in Eq. 1, where $L(x)$ and $L(y)$ are the length of the element x and y, respectively. The score is used to fill an alignment matrix as described by Needleman-Wunsch. Besides the score system, the other parameter in an alignment is the gap penalty. A gap comprises an insertion or deletion in a sequence, usually occurring from a single mutational event. SSEA method do not analyze explicitly the role of gap penalty in the alignment.

$$S(x,y) = \begin{cases} \min(L(x), L(y)) & \text{if } x = y \\ \frac{1}{2}\min(L(x), L(y)) & \text{if } (x = \{H, E\} \text{ and } y = C) \text{ or} \\ & \quad (x = C \text{ and } y = \{H, E\}) \\ 0 & \text{if } (x = H \text{ and } y = E) \text{ or} \\ & \quad (x = E \text{ and } y = H) \end{cases} \tag{1}$$

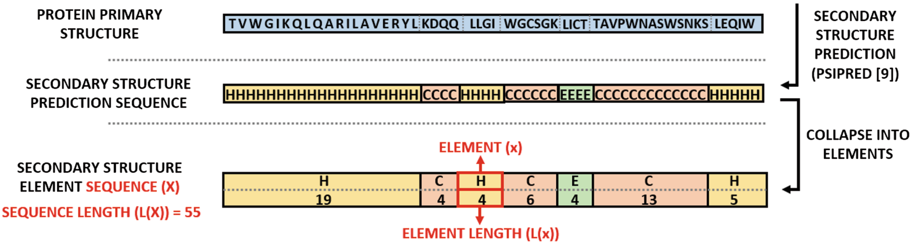
PROTEIN PRIMARY STRUCTURE
TVWGIKQLQARILAVERYL KDQQ LLGI WGCSGK LICT TAVPWNASWSNKS LEQIW
SECONDARY STRUCTURE PREDICTION (PSIPRED [9])
SECONDARY STRUCTURE PREDICTION SEQUENCE
HHHHHHHHHHHHHHHHHHH CCCC HHHH CCCCCC EEEE CCCCCCCCCCCCC HHHHH
COLLAPSE INTO ELEMENTS
ELEMENT (x)
SECONDARY STRUCTURE ELEMENT SEQUENCE (X)
SEQUENCE LENGTH (L(X)) = 55
H 19
C 4
H 4
C 6
E 4
C 13
H 5
ELEMENT LENGTH (L(x))

Fig. 2. Secondary Structure Element Alignment (SSEA) concepts. A secondary structure sequence is obtained from an amino acid sequence using Psipred [8]. The consecutive repeated characters are collapsed into elements. Each element is associated with the length of the collapsed characters. The group of ordered elements is a secondary structure element sequence. The sum of lengths of the elements equals the length of the sequence.

The final similarity score in a Needleman-Wunsch alignment corresponds to the last cell score in an alignment matrix and is normalized by the mean of the length of the two sequences. This final score $(d(X, Y))$ is between 0 and 1, the higher the score, the higher the similarity between those two proteins according to the secondary structure. Przytycka et al. proposed and applied this metric to generate a taxonomic tree through a clustering algorithm. The generated tree was compared with trees generated with methods that involve more information, and the taxonomic organization was in agreement. Almost at the same time, Xu et al. [11] used a similar measure to identify two enzymes in Archaea. They also do alignments using a dynamic programming algorithm, but do not collapse consecutive characters.

McGuffin et al. [12] proposed that the prediction of proteins secondary structure and the alignment of its elements allows to detect distant homologs in a better way than methods based on amino acid sequence. Different amino acid sequences may adopt similar tridimensional structures. The capacity to identify distant homologs from the alignment of secondary structure elements was also evaluated by Zhang et al. [13]. The identification of distant homology was accomplished through a method based on Support Vector Machines (SVM), and different metrics were compared. The classification from secondary structure alignments obtained one of the highest accuracy values. Si et al. [14] applied the method to identify proteins with a highly conserved tridimensional domain, called TIM-barrel (triose-phosphate isomerase) allowing to identify this domain in Bacillus subtilis proteome with 99% of accuracy using SVM. SSEA method was also applied successfully by Ni and Zou [15] to the prediction of outer membrane proteins from bacteria. They developed a kernel function based on the metric proposed in SSEA, capable of classifying outer membrane proteins using a method based on SVM with a 97.7% accuracy.

3 Proposed System

3.1 Development of a Similarity Measure

Our bioinformatics search is based on viral fusogens, since these are the only known fusion machineries capable of catalyzing the fusion of membranes outside cells. Because of the amount of information available, we focus on Classes I and II. Owing to the high divergence at sequence level in viral fusogens, algorithms that find similarity between amino acid sequences are frequently not enough to identify similar proteins. To solve this, we evaluate a metric capable of discriminating secondary structure signals between viral fusogens. Our task is to tune up this technique so we can evaluate it later with proteins from other biological models, as the ones described previously.

Viral fusogens are synthesized as inactive precursors (VFPs), that under certain conditions are cleaved, releasing a transmembrane protein with fusion capacity. We refer to the ectodomain as the fusogen (VF) (Fig. 3). Considering the necessity to search fusogens in proteins synthesized as precursors, our algorithm is intended to correctly align VFs with other VFs, but also with VFPs.

Our protein similarity measure is developed based on SSEA but modifies the alignment algorithm and score normalization, and explores the gap penalty incidence. When aligning the secondary structure of a VF and a VFP the algorithm will not consider the local alignment between the VF and the VFP fusogenic region, since Needleman-Wunsch algorithm computes a global alignment. For this reason, we propose to apply a local alignment algorithm, analogous to Smith-Waterman algorithm [16], which allows the correct alignment between fusogens and proteins that contain a fusogen. Thus, we perform secondary structure representations alignments (VFs and VFPs) in pairs, applying SSEA method, substituting the alignment algorithm with Smith-Waterman algorithm. Although the local alignment approach was described by Fontana et al. [17] it was not applied to a specific problem, the tool is no longer publicly available and we did not find any articles that apply this modification.

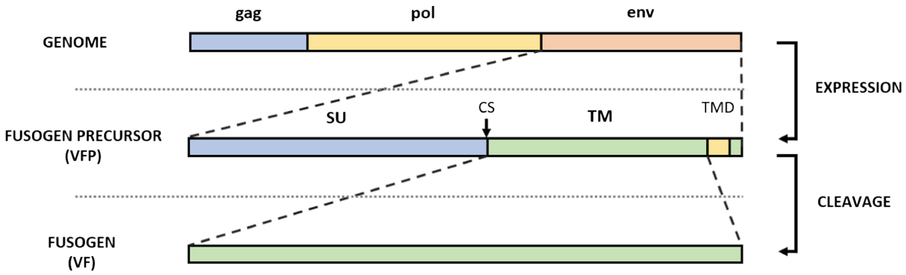

Fig. 3. Viral fusion proteins processing. In this example HIV is presented for its simplicity. Env gene is synthesized as a non-functional precursor protein (VFP) containing a surface protein (SU) and a transmembrane protein (TM). Both proteins are cleaved at the cleavage site (CS) to form an active fusion protein. The released TM is a Class I viral fusion protein. The transmembrane domain separates the protein into an intraviral domain and an ectodomain (VF). The latter carries out the fusogenic activity of the virus.

In SSEA method the final score corresponds to the last cell score in the matrix of a Needleman-Wunsch global alignment. Since we work with a local alignment algorithm, our final score is the maximum score obtained in the dynamic programming matrix as described by Smith-Waterman.

The VFs sequences length is variable between the two classes, and also inside each class. The same happens with the VFPs sequences length, so we would not expect to find a relation between the length of VFs and the respective VFPs. For this reason we could expect that the normalization method applied in SSEA would fail, since the alignment final score is divided between the mean of the pair of proteins sequences length. Thus, we propose another modification to the metric, where the final score is normalized by the mean of the aligned regions length for each pair of proteins.

3.2 Classification

Similarly to viral fusogens, we consider that protein candidates for fusion capacity in other biological models may exist as a part of a precursor. For this reason, besides evaluating the metric when classifying a group of Class I and Class II VFs, we will also evaluate the metric when classifying a group of Class I and Class II VFPs.

Another approach consists of training a One-class SVM (OC-SVM) classifier with Class I VFs and classifying a group of VFs or VFPs as Class I (positive class) or Class II (negative class). This is a first approximation to evaluate the method in order to consider its application for classifying proteins from other biological models as proteins with fusogenic capacity (positive class) and no fusogenic capacity (negative class).

4 Data Description and Experimental Results

4.1 Data Pre-processing

We obtained the amino acid sequences for the VFPs available in the public database UniProt [18]. We selected those proteins labeled as Class I viral fusion protein or Class II viral fusion protein. We obtained 27846 Class I and 1800 Class II sequences, with variable lengths from 446 to 1376 amino acids. We extracted the VF from each protein using the annotations available in UniProt. The lengths varied from 136 to 584 amino acids. From here on, we worked with the VFP and the VF in parallel.

Knowing the redundancy of sequences in UniProt, as a previous step for secondary structure prediction, the sequences were clustered with 99% identity with CD-HIT tool [19]. Thus, we obtained 1769 representative sequences of Class I viral fusogens, and 1103 Class II fusogens. We selected randomly 100 Class I VFs and 100 Class II VFs. We also selected randomly another 100 Class I VFs, 100 Class II VFs, and their corresponding 100 Class I VFPs and 100 Class II VFPs.

For these 600 sequences the secondary structure predictions were calculated with Psipred, using the HHSuite package [20]. This method considers a multiple sequence alignment for each amino acid sequence to improve the accuracy, as evolution provides a closer description of structural tendencies. Finally, we computed similarity matrices for the proposed metric. We analyzed the similarity matrices obtained for a constant gap penalty with values between 0 and -5, and chose to work with a penalty value of -1. This value maximizes the score when comparing sequences of the same class, and minimizes it when comparing different classes.

Training and Classification. SVM method has been widely used in biological sequences analysis. This method uses kernel functions, mapping the problem into a high-dimensional space. This feature allows the construction of a hyperplane that has the highest separation between two classes in the transformed space.

A distance between protein sequences was obtained from the computed similarity with the kernel [15]:

$$k(x, y) = \exp(\gamma\, d(X, Y)). \tag{2}$$

We worked with LIBSVM package [21] for Python. LIBSVM can generate a classifier from the precalculated kernel and estimate the performance.

VFs Classification Training with Two Classes. We trained a SVM classifier with a set of Class I and Class II VFs. The classification was performed with another set of Class I and Class II VFs. The performance of the classifier depends on parameters C and γ. The parameter C affects the flexibility of the classification, allowing some errors, but also penalizing. The parameter γ establishes how far the influence of a sample can reach. The best combination of C and γ was selected using a grid search with 10-fold cross-validation, with C values between 2^{-15} and 2^{15}, and γ values between 5×10^{-4} and 5×10^{2} with uniform intervals.

We selected as optimal parameters $C = 2^{-15}$ and $\gamma = 5.4 \times 10^{-1}$. For these parameters, the classification accuracy for Class I and Class II fusogens was 99.0%.

To obtain a second evaluation of the proposed metric, we classified the set of VFs using k-NN as the classification method, for 1, 3 and 7 NNs. The classification accuracy was 98.5%, 98.5% and 97.5% respectively (Table 1).

Table 1. Accuracies obtained for VFs and VFPs classification.

Classification	1-NN	3-NN	7-NN	SVM	OC-SVM
VFs	98.5	98.5	97.5	99.0	92.0
VFPs	98.5	97.5	95.5	90.5	69.5

VFs Classification Training with a Positive Class. SVM classifiers are based in training with samples belonging to two classes (e.g. positive and negative). However, in some situations there are only positive samples for training. This is the case of the problem suggested in this work, as we have a set of training proteins known to be fusogenic, and we intend to select fusogen candidates from a diverse set of proteins. Given the characteristics of the candidate proteins set and the virtually infinite variability a protein can present, it is not possible to create a representative negative samples. Schölkopf et al. [22] described the one-class classification method that allows to train a model with just positive samples.

For this reason, we trained an OC-SVM classifier with Class I VFs as positive samples. We evaluated the classification of a Class I VFs set (distinct from the training set) and a Class II VFs set. For this part, we also used LIBSVM package. The same kernel was applied to the data, and the best combination of parameters was chosen. In this case, parameter ν substitutes parameter C. The meaning of parameter ν is analogous to C meaning, but the values should be between 0 and 1. Similarly to previous part, the best combination of parameters ν and γ were selected using a grid search with 10-fold cross-validation, with ν values between 0.05 and 1, and γ values between 5×10^{-4} and 5×10^{2} with uniform intervals.

We selected as optimal parameters $\nu = 0.05$ and $\gamma = 5.4 \times 10^{-1}$. For these parameters, the classification accuracy for Class I and Class II VFs was 92.0% (Table 1).

VFPs Classification Training with Two Classes. In order to evaluate the modified algorithm performance for local alignments, the SVM classifier was evaluated classifying a group of VFPs. It was trained with the same two classes used previously. The classification accuracy for Class I and Class II VFPs was 90.5%. The k-NN classification accuracy for 1, 3 and 7 NNs was 98.5%, 97.5% and 95.5% respectively (Table 1).

VFPs Classification Training with a Positive Class. The classification of Class I and Class II VFPs when training only with Class I VFs resulted in an accuracy of 69.5% (Table 1).

5 Conclusions and Future Work

The developed metric, based on SSEA allowed the satisfactory classification of VFs using the three proposed methods (k-NN, SVM y OC-SVM). The classification of VFPs using k-NN gave a similar accuracy as the obtained for VFs classification. We also obtained an acceptable accuracy when classifying VFPs using a SVM model. However, the performance is reduced considerably when classifying VFPs using an OC-SVM model. It is clear that a reduction of the accuracy is expected since this is the most challenging case where the classifier is trained with VFs and tested with VFPs, and the reduction for the two classes SVM is already greatly reduced. However, the reduction is too abrupt, and further analysis of the dissimilarities between classes would help to understand the reasons of this reduction. In spite of the SVM and OC-SVM classification results, the metric by itself appears to accomplish the objective according to the results obtained for 1-NN classification.

This work was performed on a reduced set, selected randomly from an original set of VFs and VFPs obtained from UniProt. We discarded a significant subset of sequences from the original set as those sequences did not have annotations for the cleavage of VFPs. The first step when reviewing the metric should be to expand the set.

On the other hand, we propose to work in detail on the influence of gap penalties in the metric. Evaluations not presented in this work showed that gap penalty value does not have influence on the performance of k-NN classifiers, but does have influence on SVM and OC-SVM classifiers. In this work, we used a constant gap system, so the gap penalty value is always the same. It would be interesting to evaluate the performance of the metric when working with a linear gap system (dependent on the length of the gap) or with affine gap penalty, in which the gap opening is penalized differently than gap extension.

The set-up of this method could make possible the identification of unknown viral fusogens from the genome or proteome of enveloped viruses. It could also be used to identify proteins with fusogenic capacity in different biological models.

Acknowledgment. This work was partially supported by Agencia Nacional de Investigación e Innovación (DCI-ALA/2011/023-502 "Contrato de apoyo a las políticas de innovación y cohesión territorial"). We would also like to include the double affiliation of Federico Lecumberry (Laboratorio de Procesamiento de Señales, Institut Pasteur de Montevideo, Uruguay).

References

1. Rochlin, K., Yu, S., Roy, S., Baylies, M.K.: Myoblast fusion: when it takes more to make one. Dev. Biol. **341**, 66–83 (2010)

2. Primakoff, P., Myles, D.G.: Penetration, adhesion, and fusion in mammalian sperm-egg interaction. Science **296**, 2183–2185 (2002)
3. van der Pol, E., Bing, A.N., Harrison, P., Sturk, A., Nieuwland, R.: Classification, functions, and clinical relevance of extracellular vesicles. Pharmacol. Rev. **64**, 676–705 (2012)
4. Harrison, S.C.: Viral membrane fusion. Nat. Struct. Mol. Biol. **15**, 690–698 (2009)
5. Kielian, M., Rey, F.A.: Virus membrane-fusion proteins: more than one way to make a hairpin. Nat. Rev. Microbiol. **4**, 67–76 (2006)
6. Perez-Vargas, J., Krey, T., Valansi, C., Avinoam, O., Haouz, A., Jamin, M., Raveh-Barak, H., Podbilewicz, B., Rey, F.A.: Structural basis of eukaryotic cell-cell fusion. Cell **157**, 407–419 (2014)
7. Cuff, J.A., Clamp, M.E., Siddiqui, A.S., Finlay, M., Barton, G.J.: JPred: a consensus secondary structure prediction server. Bioinformatics **14**, 892–893 (1998)
8. McGuffin, L.J., Bryson, K., Jones, D.T.: The PSIPRED protein structure prediction server. Bioinformatics **16**, 404–405 (2000)
9. Przytycka, T., Aurora, R., Rose, G.D.: A protein taxonomy based on secondary structure. Nat. Struct. Biol. **6**, 672–682 (1999)
10. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**, 443–453 (1970)
11. Xu, H., Aurora, R., Rose, G.D., White, R.H.: Identifying two ancient enzymes in Archaea using predicted secondary structure alignment. Nat. Struct. Biol. **6**, 750–754 (1999)
12. McGuffin, L.J., Jones, D.T.: Targeting novel folds for structural genomics. Proteins **48**, 44–52 (2002)
13. Zhang, Z., Kochhar, S., Grigorov, M.G.: Descriptor-based protein remote homology identification. Protein Sci. **14**, 431–444 (2005)
14. Si, J.N., Yan, R.X., Wang, C., Zhang, Z., Su, X.D.: TIM-finder: a new method for identifying TIM-barrel proteins. BMC Struct. Biol. **9**, 73 (2009)
15. Ni, Q., Zou, L.: Accurate discrimination of outer membrane proteins using secondary structure element alignment and support vector machine. J. Bioinform. Comput. Biol. **12**, 1450003-1–1450003-12 (2014)
16. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Mol. Biol. **147**, 195–197 (1981)
17. Fontana, P., Bindewald, E., Toppo, S., Velasco, R., Valle, G., Tosatto, S.C.: The SSEA server for protein secondary structure alignment. Bioinformatics **21**, 393–395 (2005)
18. UniProt Consortium: UniProt: a hub for protein information. Nucleic Acids Res. **43**, D204–D212 (2015)
19. Li, W., Godzik, A.: Cd-Hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics **22**, 1658–1659 (2006)
20. Soding, J., Biegert, A., Lupas, A.N.: The HHpred interactive server for protein homology detection and structure prediction. Nucleic Acids Res. **33**, W244–W248 (2005)
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011)
22. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support vector method for novelty detection. In: NIPS, vol. 12, pp. 582–588 (1999)