User documentation for running the Peptide Classifier from Sequence and Structure (PCSS) web server
9-15-2010
Please direct all feedback and questions to dbarkan@salilab.org

Document Outline

1 Introduction

This server implements the algorithm described in the manuscript "Prediction of protease substrates using sequence and structure features" by Barkan, Hostetter *et al*, Bioinformatics 2010.

This document is an overview to using the service. More information can be found by clicking the "help" link on the front page. Additionally, questions can be directed to the site administrator and the authors by following the "contact" link on the first page.

2 Overview

2.1 Biological Relevance

Often in nature, protein-protein recognition is mediated by small peptide regions on one protein binding to a pocket or groove on another protein. This is the case in scaffolding domains such as PDZ and SH3 which recognize peptides six to ten residues in length, in protease-substrate specificity in which the substrate peptide associates with the protease active site before catalysis, and in many other systems as well. This recognition is mediated by the sequence of the peptide and the structural context in which it falls in its protein. It is often of interest for researchers to identify these peptides; for example, predicting a peptide that is cleaved by a protease, thus affecting the larger protein substrate, can lead to hypotheses as to the effect of this cleavage on modified substrate function or lack thereof.

To aid in this prediction effort, this webserver implements a protocol that lets the user provide positive and negative examples of these peptides. From this training data, a statistical model is generated that can

then be used in the server to search for similar peptides in other protein sequences. Full details are described in the accompanying manuscript.

2.2 Server Usage

There are two modes in which to use the server, Training mode or Application mode. Only one mode is run at a time; often the user will want to run it in Training mode to train a model, wait for completion, and subsequently run it in Application mode to apply the model to make new predictions. Alternatively, the user can run it in Application mode only to use one of the provided models.

Training Mode

In training mode, the user can upload a set of peptides that are known positive examples of the peptide motif in nature. For example, these can be peptides recognized by MHC Class II proteins, as demonstrated in literature or in experiments conducted by the user. We refer to these as the Positive set. Additionally, the user provides a set of Negative peptides, that is, those that are known NOT to be involved in the recognition process. For each of these training peptides, the user specifies the protein (referenced through its accession in the Uniprot database), the position of the peptide in the protein, the peptide sequence itself, and whether the peptide is a positive or negative example (the exact file format is specified in Section 3).

Upon submission, the sequences are validated through the ModBase database, and then secondary structure, solvent accessibility, and disorder annotation is generated for the peptides. In cases where a solved structure or good quality comparative model is available for the protein, secondary structure and solvent accessibility is assessed with the DSSP algorithm. In cases where a structure is not available, as well as with all disorder predictions, the features are evaluated using sequence-based prediction methods. These structural features, along with the peptide sequence itself, are encoded as input for a Support Vector Machine (SVM) learning model. A percentage of the input is not involved in the training; rather, it is set aside and used as a test set. After training, the model is applied to the test set to score the peptides contained therein, and its success in discriminating positives from negatives is evaluated. This process is repeated and the results are averaged.

Upon completion of the training, the server returns the results of the training, including the false and true positive rates in a format that can used to construct a Receiver-Operator Characteristic (ROC) curve, and the SVM model itself. The ROC plot can be examined to determine if there is enough information in the training data to use the model to make predictions. The SVM model can be used in subsequent runs of the server in Application mode.

Please note that currently we are implementing this functionality of reusing user-generated SVM models in application mode and it will be available in the near future.

Application Mode

In Application mode, the user uploads a set of proteins to be evaluated using the SVM model to search for high-scoring peptides that match the motifs in the training set. The peptides can either be listed explicitly, or the user can simply provide Uniprot accessions in which to search for peptides, along with certain restrictions on the peptide sequence to narrow the search down. Upon submission, a similar process occurs to that in training mode, where the proteins are validated and then annotated with the same

structural features. The model is used to score the peptides, and the results are returned by the server to the user.

3. User Input

Following is a description of each option on the server front page. Please note that the server mode (Training vs Application) is selected with the **Server Mode** option, but fields for Training and Application modes are both included on the same page. Each is submitted using the submit button on the bottom of the page. Entries in fields corresponding to the other mode (that is, the one not selected with **Server Mode**) are ignored.

3.1 Global options (set in both Training and Application mode).

- **Server Mode:** Use this option to select whether to train a new predictive model on a set of known positives and known negatives that you provide ("Training") or to use an existing model to search a set of proteins to look for high scoring sites of interest ("Application").

- **E-mail Address:** Upon job completion, an email will be sent to this address to notify you, with a link to the results

- **Best Protein Structure Model Criteria:** This option allows the user to choose which criteria will be used to select the best quality comparative model when one is available to evaluate secondary structure and solvent accessiblity for a peptide in a protein. Often there will be more than one model generated for the protein, but only one is used to evaluate these structure features, chosen according to this criteria. The following options are available:
    - **Model Score:** A general score for the reliability of a model, as described in the ModBase documentation.
    - **Model coverage:** Fraction of the target protein sequence for which a comparative model was generated.
    - **Predicted Native Overlap:** Fraction of α-carbons in the model that are predicted to be within 3.5 angstroms of the native state.

  The algorithm for predicting the native overlap is described in this study. It was shown to outperform the Model Score metric in evaluating comparative model quality, but the two are generally correlated, and users who are familiar with ModBase may wish to use Model Score. Model coverage is a useful evaluation metric when a user wishes to evaluate a model that has a larger fraction of the target included, for example when only a small portion of the target is included in the model but receives a very high score.

- **Select Error Handling Mode:** Error handling allows the user to specify what to do if a Uniprot accession is not found in ModBase. There are two options:
    - **Quit:** This will inform the user of an accession error immediately and allow immediately and allow them to correct it.
    - **Ignore:** This will continue processing the job, and a message will be displayed in the results file for the missing accession.

If searching through a small number of accessions for high scoring peptides, choosing the **Quit** option is suggested, as the user will not have to wait for the job to complete before discovering there is a problem. However, if a large number of accessions are uploaded, it is probably more beneficial to choose the **Ignore** option and just receive a list of all accessions that were missed upon job completion.

3.2 Training Input

- **Upload Training Peptide File:** In Training mode, the uploaded file is a list of peptides found in proteins that are known to be either postive or negative examples in a system involving specificity in protein-peptide recognition. For example, in studying which peptide sequences are recognized by a protease, the file could include a list of peptides known to be cleaved by the protease (the positives) and other peptides known *not* to be cleaved (the negatives). The file can include multiple peptides from one protein.

  Each line in the file should include one peptide. The peptide is specified as such:

  Accession Start_Position Sequence Classification

  Accession: The Uniprot accession of the protein containing the training peptide
  Start_Position: The position in the sequence representing the first residue of the peptide (Position counting begins with 1)
  Sequence: The amino acid residue sequence of the peptide
  Classification: This should be set to "Positive" or "Negative", reflecting whether the peptide is a positive or negative example of specificity

  Entries on each line are separated by a space or tab. Uniprot accessions are mapped to IDs in ModBase, a database of comparative protein structure models. ModBase continually updates this mapping. Occasionally a Uniprot accession will not be found in ModBase. This could be due to ModBase not having processed this sequence yet in its mapping, or that the accession is outdated or not the primary accession for the protein. Note that accessions in Uniprot contain six characters and are distinct from Uniprot entry names, which are up to 11 characters and contain an '_'.

  Additionally, all peptides must be the same length. The provided peptide start position and peptide sequence must match the residue sequence for the Uniprot accession in ModBase. The number of negatives in the file must be greater than or equal to the number of positives (if this is not the case in your datastet, you can simply reverse your definition of Negative and Positive for each peptide.)

  An example follows of one line in the training file:

  O75791 11 SGEDELSF Positive

  This represents the peptide SGEDELSF, found in the protein specified by Uniprot accession O75791, starting at position 11. The protein will be considered a known positive in the training set

- **Training Iterations:** In Training mode, use this option to select how many Jackknifing iterations to run. Each iteration trains on a portion of the known positives and negatives listed in the Peptide Training File and tests on another fraction (specified by the option "Jackknife Fraction") of these known positives and negatives. It is suggested that multiple iterations of the Jackknifing procedure are run to avoid biasing the results to which peptides are chosen to be in the test set. The maximum number of iterations that can be run is 1000. The results of the procedure are averaged to produce final results.

- **Jackknife Fraction:** In Training mode, use this option to select what percentage of positive peptides in the Training file are set aside for testing; the remaining set of positives will be used for training. An equal number of negatives will be included in the training set, and the remaining negatives will be used in testing.

  Thus, if you have 100 positives and 200 negatives in your training file, and specify .1 as the jackknife fraction, there will be 90 positives and 90 negatives in the training set, 10 positives in the test set, and 110 negatives in the test set.

3.3 Application Input

- **Upload Application Target Peptide File:** In application mode, the uploaded file is a list of uniprot accessions, one per line. The predictive model will search for high scoring peptides in the protein sequences for these accessions.

  These accessions are mapped to IDs in ModBase, a database of comparative protein structure models. ModBase conitnually updates this mapping. Occasionally a Uniprot accession will not be found in ModBase. This could be due to ModBase not having processed this sequence yet in its mapping, or that the accession is outdated or not the primary accession for the protein. Note that accessions in Uniprot contain six characters and are distinct from Uniprot entry names, which are up to 11 characters and contain an '_'.

  Note that only human protein sequences currently have structural features calculated for them from solved structures and comparative models in this framework. Sequences from other species will always be evaluated with sequence-based structural prediction algorithms. More organisms may be represented as this service is developed. If you are interested in training or fully testing a specific species other than human, please contact the administrators.

- **Upload Peptide Specifier File (Optional):** In application mode, the uploaded file allows the user to specify restrictions on the peptides they wish to score. These take the form of allowing only certain residues to be present at different positions in the peptide. This reduces the number of peptides scored and output, and allows the user to incorporate knowledge of the system into predictions. A demonstrated is given by the GrB and caspase systems, where these proteases recognize peptides containing an aspartic acid at the P1 site. The provided GrB and caspase predictive models were optimized on these peptides, so it follows that when applying these models to search for high-scoring peptides, the user should restrict peptides to have the same property.

The format of the specifier file is simple. Each position to restrict is designated with one line in the file. The line should begin with a number representing the position in the peptide. This is followed by a space separated list of residues that should **not** be present in that position. If there are no restrictions on a position in the peptide, then there does not need to be a line specifying that position. Position counting is 1-based.

An example follows:

```
1 E D
4 A C E F G H I K L M N P Q R S T V W Y
8 W
```

This file will tell the service to only score peptides that don't have an acidic residue in the first position, that only contain Asp in the fourth position, and that don't have Trp in the 8th position.

Note the number of the maximum position to restrict should not exceed the length of the peptides on which the predictive model was trained. For the provided GrB and Caspase predictors, this length is 8, representing the peptide from P4 to P4'.

- **Predictive Model to Apply:** This lets the user select which predictive model to use to score the peptides found in the accessions. Each model has been trained on a different set of known positives and negatives and is optimized for sequence and structure features in that set. Currently the available models include one to predict which peptides will be recognized by Granzyme B and another for Caspase peptides.

4. Tips for ensuring a good quality server run

4.1 Training

- **Size of training set -** The number of positives that are included in the training set will greatly affect the predictive ability of the resulting model. It is suggested that greater than 100 positives should be included for optimal results.

- **Working with small training sets -** If there are not many examples of positives available, try varying the jackknife fraction from 0.10 to 0.50 in multiple runs of the server in training mode. When it is low, there are more examples in the training set, which allows for a better model to be built and applied to the test set. When it is high, there are more examples in the test set, which allows for a more robust ROC plot. If a training set produces a good ROC plot across different jackknife fractions, one can be confident using it to make predictions even if it is small.

- **Specifying negatives -** Often the question will come up of how to choose negatives for inclusion. These must be peptides of the same length as the positives that are known not to bind to the protein of interest. One way to select these is to examine the protein that includes the positives, and find other peptides in this same protein that are not the positives, which will be the majority of peptides. It is also useful to have the negatives be similar in sequence to the positives; if they are very different, the model will be built based on the extremely common features across all positives

and make predictions based on these features. For example, in the paper describing the algorithm, the positives were peptides cleaved by GrB and caspases with Asp in the fourth position, and the negatives were all peptides in the same protein substrates that had Asp in the fourth position, but were not cleaved (See Suppl. Figure 1b). If instead, negatives had been any peptide with arbitrary sequences, then the model would have erroneously predicted all subsequent peptides with Asp in the fourth position to be a positive. The true predictive value of the server lies in the peptide positions that have degenerate residue preferences, usually by also incorporating structuree features.

- **Interpreting results -** Regardless of the size of the training set, a ROC plot is produced and should be examined to determine if it is able to distinguish positives from negatives appropriately to have confidence in future predictions made by the model. The TPRs can be plotted against the FPRs to produce a visual representation of the ROC plot, which spans from (0,0) to (1,1). A good predictor will have a critical point of at least 0.8 TPR at a 0.2 FPR (the critical point is where the curve crosses the line spanning (1,0) to (0,1). This implies that a score at the critical point will evaluate predictions at a 20% false positive rate, without sacrificing much coverage of true positives (taking into consideration the actual number of positives and negatives in the training set). A training set that contains no information will have a ROC plot falling exactly on the line from (0, 0) to (1,1); this is the same as guessing.

4.2 Application

Dataset size – While there are no restrictions on the number of proteins that can be evaluated in Application mode, we suggest evaluating a few thousand at a time at the most. This reduces the load on the server and ensures that a node going down on the cluster or other uncommon errors will not crash a larger run. Large datasets can easily be broken up into smaller sets and submitted in succession to account for this.

Job submission – Especially when submitting large datasets, please allow some time for the job to submit, and don't click the submit button more than once. During this time, many database queries are being made, which can take some time.

Interpreting results – The output file will produce a score for each peptide parsed from the protein or specified by the user. Scores are generally not meaningful, but are used to determine where the peptide would have been fallen in the ROC plot, in terms of its FPR and TPR. Thus, if a peptide gets a score that gives it an FPR of 0.15 and TPR of 0.83, that means that 83% of the positives in the benchmark set scored better than the peptide, and 15% of the negatives scored better. The best peptide scores are those with very low false positive rates (less than 0.05), and if experimental cost and time are considerations, then these should be the ones focused on in validation studies.

5. Output

The results of both server runs are output to files which are available for downloading at a link specified in the email sent when a run is finished.

5.1 Training output

Training Results

In training mode, the server returns a file indicating the predictive ability of the model. This is a reflection of how much information is contained in the positive examples to distinguish them from the negatives. As noted above, the server trains on the majority of the input peptides and sets aside some for testing.

The test set peptides are ranked by score, from best to worst. The analysis traverses the ranking, starting at the top. For each peptide entry in the ranking, the number of respective positives and negatives falling at or above the score for the entry are counted and output as a percentage of the total number of positives and negatives in the test set. Thus, if the test set included 100 positive and 100 negatives, and after the 10th best-scoring peptide there had been 8 positives and 2 negatives counted, the output would be .08 true positive rate (TPR) and .02 false positive rate (FPR). This continues until the full rank has been traversed, so that the last entry will be a TPR and FPR each of 1.0.

Since the training and testing procedure is repeated over a number of iterations, the output in the file is the average FPR at each TPR. Collectively, the FPR / TPR pairs can be used to generate a ROC curve for final assessment of the model by plotting column 2 against column 1.

The following columns are included in the results file:

- **Columns 1 and 2:** False Positive Rate (fpr) and True Positive Rate (tpr), as described above.

- **Column 3:** The average SVM score for the positive peptide at this rank. The value of the SVM score is of little consequence; it is only used to compare one peptide to another in ranking, and later can be used to compare the score of a target peptide to the benchmark set.

- **Column 4:** The standard deviation of the false positive rate at this level over the number of iterations specified by the user.

5.2 Application Results

In Application mode, the results file includes one line for each peptide scored. The following fields are written to the file (***** indicates this field is only written if there is a good quality model for the protein that contains the peptide):

- **Uniprot Accession** – The Uniprot Accession of the protein, as supplied by the user.
- **SVM Score –** the final SVM score assigned by the model to the peptide.
- **TPR –** the true positive rate that the score for this peptide would have been assigned in the benchmark set ROC plot (can also be read as the fraction of benchmark set positives that received a better score than this peptide)
- **FPR** – the false positive rate that the score for this peptide would have been assigned in the benchmark set ROC plot (can also be read as the fraction of benchmark set negatives that received a better score than this peptide)
- **Classification –** Whether the peptide is a positive example of binding or negative, as assigned by the user (Training Mode)
- **Peptide Sequence** – the full amino acid residue sequence of the peptide
- **Peptide Start** – the position in the protein where the peptide starts
- **Peptide End –** the position in the protein where the peptide ends

- **Peptide Secondary Structure Type –** The DSSP predictions for whether the peptide residues are on a loop, helix, or sheet
- **Peptide Accessibility Prediction –** Whether the peptide residues are exposed or buried ('exposed' indicates that a fraction of the residue greater than 0.16 is exposed to solvent)
- **Disopred Prediction** – Whether the residues in the peptide are ordered or disordered, as predicted by Disopred
- **PSIPRED Prediction –** The PSIPRED predictions for whether the peptide residues are on a loop, helix, or sheet
- **Model URL* –** A link to view the model in modbase
- **Total Models For Sequence* –** Number of models that exist in ModBase for this sequence, regardless of whether they include the region containing the peptide
- **Models Containing Peptide* –** Number of models that exist in ModBase for this sequence that include the full peptide
- **Dataset*** – The ModPipe run that was used to generate the model
- **Target Start*** – The position in the protein of the first residue included in the model
- **Target End*** – the position in the protein of the last residue included in the model
- **Model Score* –** the Model Score in ModBase of the model
- **Dope Score* –** the DOPE score assigned to the model
- **TSVMod Native Overlap*** – the predicted number of α-carbon atoms that are within 3.5 angstroms of their positions in the native state, as generated by TSVMod
- **TSVMod Method*** – The method that TSVMod used to make its predictions
- **Template Sequence Identity*** – Sequence identity between the modeled target protein and its template
- **Model Coverage* –** Fraction of the target protein that was modeled
- **Loop Length* –** If the peptide is completely on a loop region of a modeled protein, the number of amino acids on the full loop
- **Template PDB ID* –** The PDB ID of the template that was used to create the model
- **Peptide Similarity To Template* –** Whether the template residues that align to the peptide are the same as the peptide residues themselves, or different
- **Corresponding Sequence in Template* –** The sequence of the residues in the template that align to the peptide
- **Peptide Structure Values*** – The encoding of the DSSP structure prediction to integers for SVM application
- **Peptide Predicted Accessibility Fraction* –** The DSSP predictions for the fraction of the residue that is exposed to solvent
- **Disopred Scores –** The raw disorder score that is assigned to the residues in the peptide by Disopred
- **PSIPRED Scores –** The raw structure score that is assigned to residues in the peptide by PSIPRED
- **Sequence ID –** The internal modbase sequence ID for the protein
- **Model ID* –** The internal modbase model ID for the model
- **Protein Name** – Name of the protein
- **Errors –** Whether any errors occurred in calculating peptide features