**Supplementary Documentation of KSP-PUEL: a positive-unlabeled ensemble learning tool for kinase substrate prediction from dynamic phosphoproteomics data**

Pengyi Yang 1,2\*, Sean J. Humphrey 3, David E. James 4, Yee Hwa Yang 5, Raja Jothi 1,2\*

1Systems Biology Section, 2Epigenetics & Stem Cell Biology Laboratory, National Institute of Environmental Health Sciences, National Institutes of Health, RTP, NC 27709, USA. 3Department of Proteomics and Signal Transduction, Max-Planck-Institute of Biochemistry, Martinsried, Germany. 4Charles Perkins Centre, School of Molecular Bioscience, Sydney Medical School and 5School of Mathematics and Statistics, The University of Sydney, NSW 2006, Australia.

**Content**

*1. Introduction*

*2. Getting Started*

*3. Preparing Data Files for KSP-PUEL*

*4. Loading Data*

*5. Running Analyses*

*6. Interpreting KSP-PUEL Results*

*7. Step by Step Examples*

**1. Introduction**

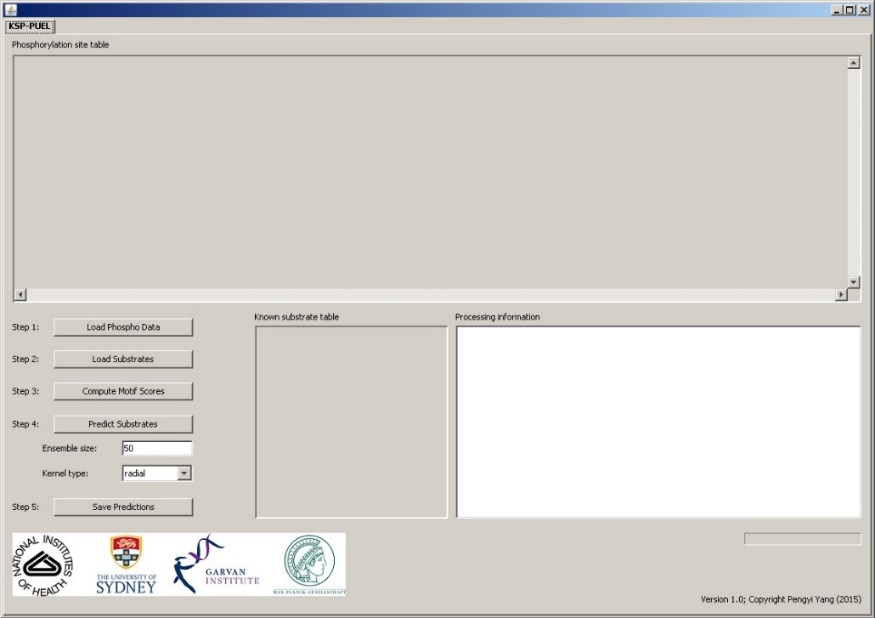
KSP-PUEL (kinase substrate prediction using positive-unlabeled ensemble learning) is designed for integrating sequence motif information and mass spectrometry phosphoproteomics data to predict kinase substrate of protein phosphorylation. KSP-PUEL is either available as an easy-to-use Java graphical user interface (GUI) or an R package. Here we provide the user guide for the GUI application. The description of R package can be found on the project homepage (https://github.com/PengyiYang/KSP-PUEL).

**2. Getting Started**

To start the GUI application of KSP-PUEL, download the jar package “KspPuel.jar” from the project homepage (https://github.com/PengyiYang/KSP-PUEL) and double click on the jar package icon. The other way to start the GUI is to open a console, change to the folder that contains the jar package, and execute the following command:

java -jar KspPuel.jar [or] java -jar -Xmx2G KspPuel.jar

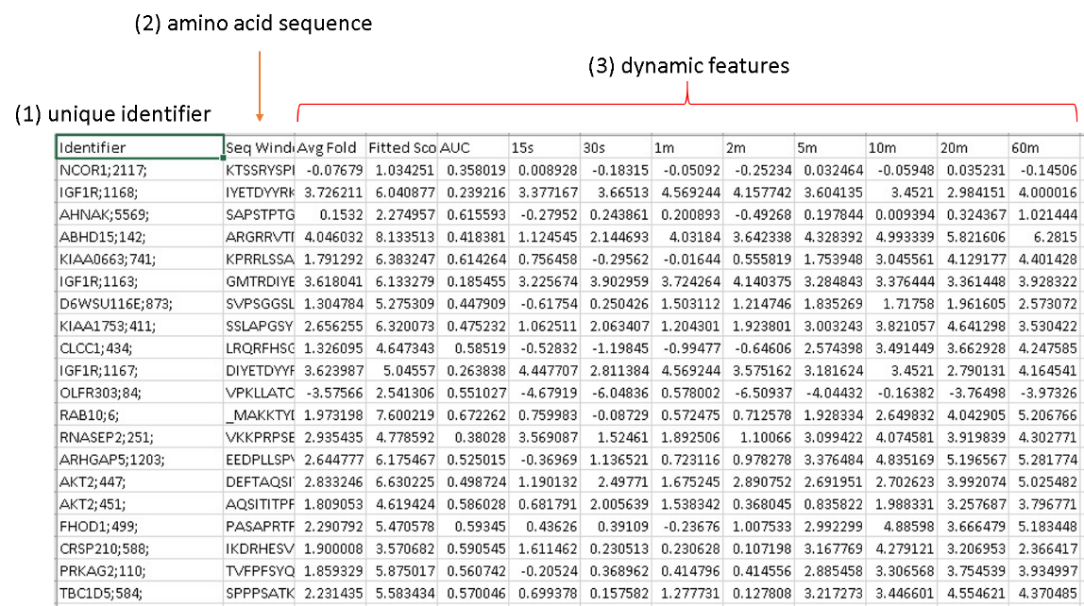
java runtime environment (jre 1.7 or above) is required for launching the application and both ways will bring up the GUI that looks similar as follow depending on the operating system one is using:



**3. Preparing Data Files for KSP-PUEL**

When using KSP-PUEL, the user will need to supply two files: (1) a *phosphoproteomics dataset file*, and (2) a *known substrate list*. The *phosphoproteomics dataset file* should have the following three components:

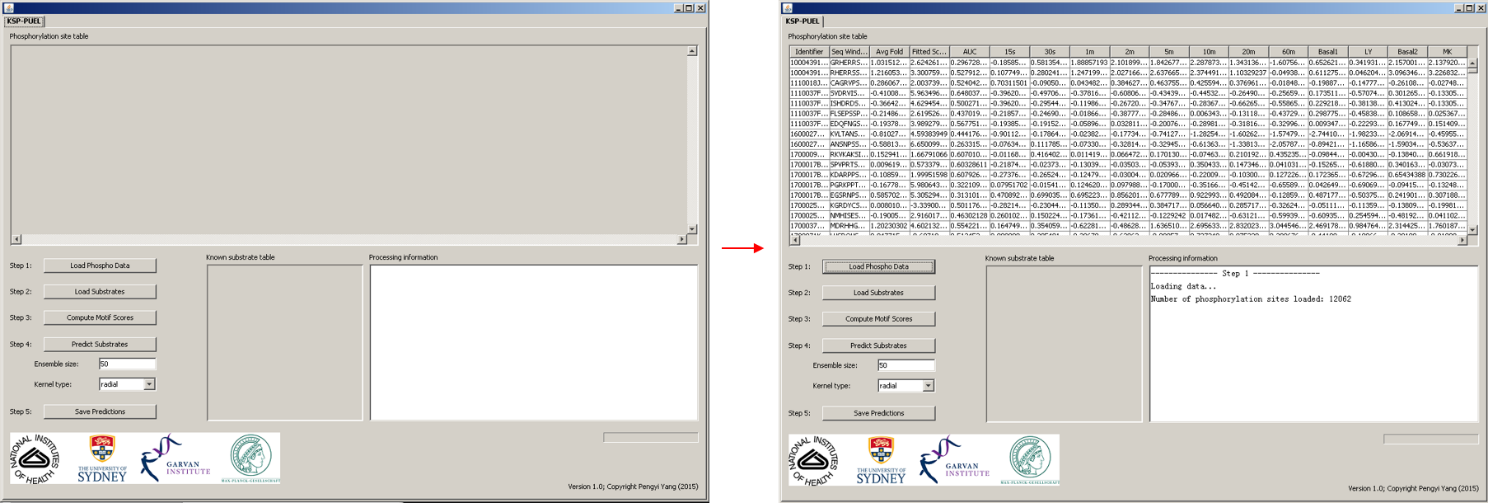
* The first column contains the unique identifiers that correspond to each phosphorylation site.
* Second, the second column contains the amino acid sequences with the middle point correspond to the phosphorylation sites. The length of the amino acid sequences can be different in different datasets but has to be consistent in a single dataset. For the positions that are out of the boundary of the sequence, use “\_” to represent that position (e.g. QFSYSASGTA\_\_\_ or \_\_MAKAYDHLFKL).
* One or more dynamic feature(s) extracted from the phosphoproteomics experiments should be added starting from column three. For time course phosphoproteomics data, these can be the quantitation of each phosphorylation site at each time point from the temporal phosphoproteomics experiments as well as secondary features extracted from the temporal profiles. For other phosphoproteomics data, these can be the quantitation of each phosphorylation site with respect to various experimental treatments. An example of the *phosphoproteomics dataset file* for KSP-PUEL is as follow (note that this example file “InsulinPhospho.txt” is available from https://github.com/PengyiYang/KSP-PUEL/tree/master/Example%20datasets/Humphrey):

****

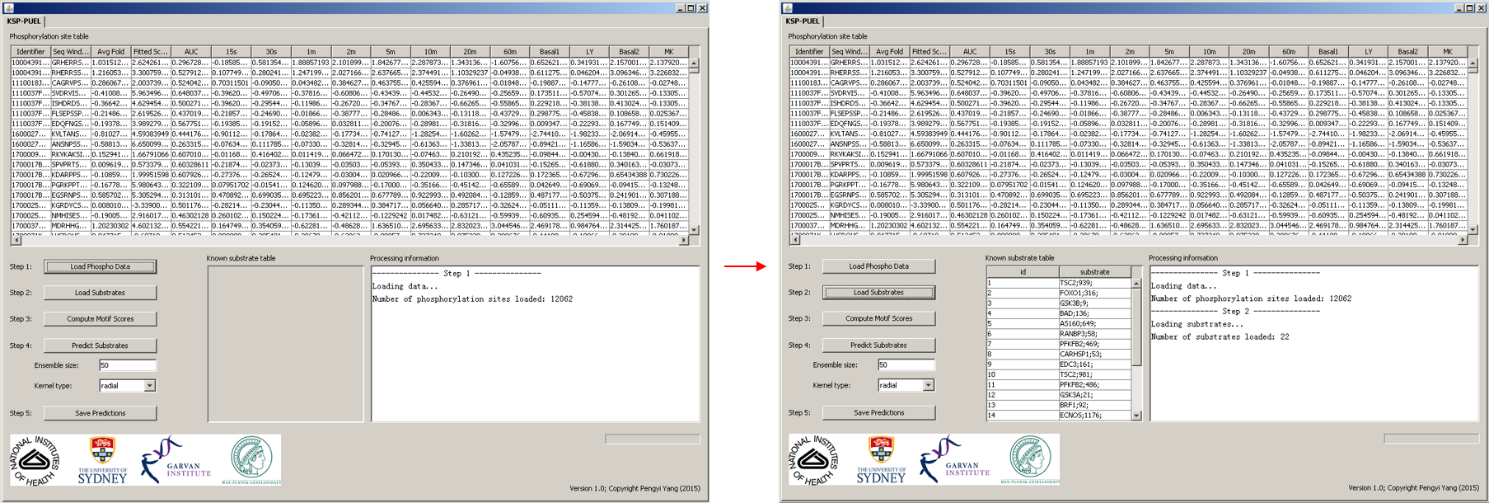
The *known substrate list* contains the substrates of a given kinase (e.g. known substrates of Akt) that will be used for training the ensemble model. This is a single column file with each row contains a unique identifier that map to the phosphorylation site in the *phosphoproteomics dataset file*. Note that this list greatly impact on the prediction results as the number and the confidence of the known substrates provided in the list are key factors for creating an accurate model from which novel substrates are predicted.

**4. Loading Data**

To use KSP-PUEL for prediction, we first load the *phosphoproteomics dataset file* by clicking “Load Phospho Data” button and choose the data file as described in Section 3. This will fill the dataset to the “Phosphorylation site table” as shown below:

****

Second, we load the *known substrates list* by clicking “Load Substrates” button and choose the known substrate list as described in Section 3. This will fill the “Known substrate table” as shown below:

****

**5. Running Analyses**

After supplying the required files, we are ready to run the analyses. First, we need to calculate a position-specific scoring matrix (PSSM) for learning the motif features from the amino acid sequences of the known substrates. The PSSM is then used to score all phosphorylation sites present in the dataset. This is done by a single click of “Compute Motif Scores” button. Next, we are ready for the final prediction. The ensemble size is defaulted at 50 and the kernel type of the SVM radial. These parameters are recommended but can be changed by users. After selecting these parameters, clicking “Predict Substrates” button to start the model training and substrate prediction procedure. The amount of time required for prediction varies depending on the size of the datasets, the learning parameters (e.g. ensemble size), and the hardware of the computer used for executing the Java application. When the prediction is done, the finial prediction score will be added to the last column of the “phosphorylation site table”.

**6. Interpreting KSP-PUEL Results**

The prediction results are added to the last column of the “phosphorylation site table”. These are probabilities estimated by the ensemble model for each phosphorylation site being a substrate of a given kinase whose known substrates are used for model training. The higher the prediction score, the more likely a given phosphorylation site is a substrate of a kinase of interest. Ideally, known substrates as well as other unannotated phosphorylation sites would be ranked at the top of the table when sorted by the prediction scores. Unannotated phosphorylation sites that have similar prediction scores as the known substrates could be selected as novel candidates for future validation.

Finally, the “phosphorylation site table” can be exported by clicking the “Save Predictions” button. Thus, the predictions can be analyzed further in excel or other types of software.

**7. Step by Step Examples**

A few example datasets are provided for demonstrating the tool. Here, we lead the users through an example analysis.

We assume that the Java runtime environment is properly installed and the “KspPuel.jar” application can be launched (see Section 2 about how to launch the application). First, on the project homepage (<https://github.com/PengyiYang/KSP-PUEL>), download the folder “Example datasets/Humphrey” that contains the files “InsulinPhospho.txt”, “Akt\_substrates.txt”, and “mTOR\_substrates.txt”.

Step 1: click “Load Phospho Data” button and select “InsulinPhospho.txt” file. This will load the phosphoproteomics data.

Step 2: click “Load Substrates” button and select “Akt\_substrates.txt” file for predicting Akt substrates or “mTOR\_substrates.txt” file for predicting mTOR substrates.

Step 3: click “Compute Motif Score” button to generate the PSSM feature for each phosphorylation site.

Step 4: click “Predict Substrates” button to learning the prediction model and predicting substrates.

Step 5: click “Save Predictions” to export the prediction results for further analysis.