**IKAP Tutorial**

In the following, a step-by-step-description of the application of IKAP to the mitosis dataset is given. This tutorial covers IKAP part 1 and 2. Part 3 is dependent on the user´s choice of database and motif validation datasets, but can be easily applied using the function descriptions included in the supplement.

We start with the dataset *data-mit* (part of the supplemental material, figure 1) and kinase-motif-information from the PhosphoSitePlus database (*PSP)* (figure 2)*,* which can be downloaded at <http://www.phosphosite.org/staticDownloads.do>. Beware that the database is frequently updated. Thus, you may obtain slightly different results for this dataset.

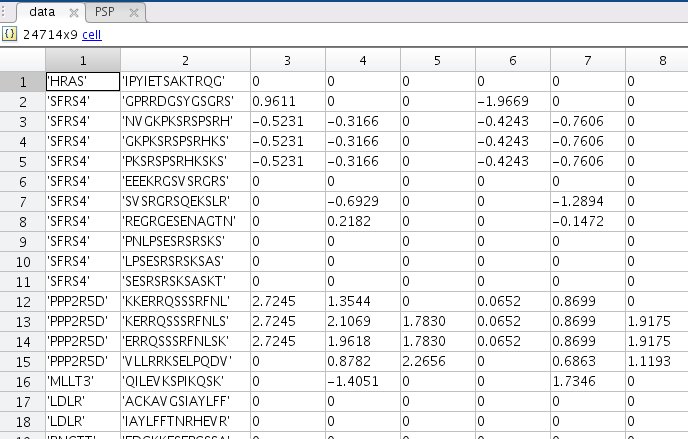


Fig. 1: Mitosis dataset. The first column should contain the protein identifiers, the second column the detected sequences with the modified amino acid in the middle, followed by an arbitrary number of data columns (6 in this case).



Fig. 2: PSP data (downloaded from PSP homepage), should be reduced to one column with kinase names and one with motifs.

**Step 1-1:**

[data]=SearchMotifs(data,PSP);

Adds an extra column to the data containing the kinases known to phosphorylate the respective motif (figure 3).

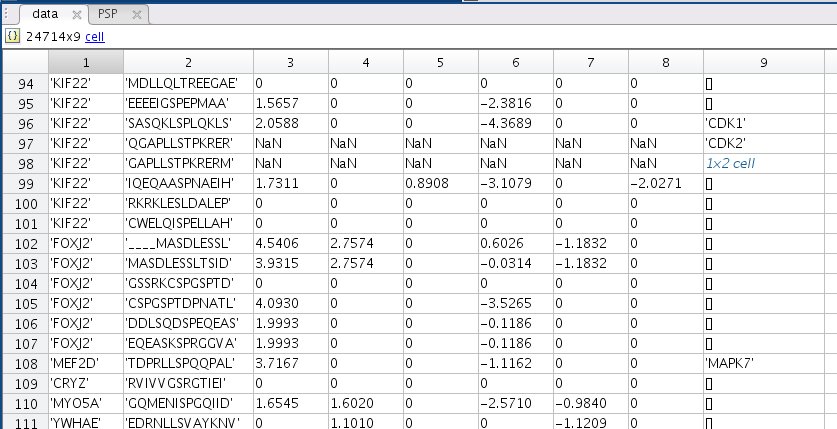


Fig. 3: Data with extra column for kinases

**Step 1-2:**

[data\_red,kin]=MakeKin(data,kcolumn,kproteome);

Reduces the dataset to those phosphosites for which a kinase is known (*data\_red,* figure 4) and generates a list of these kinases (*kin,* figure 5). *kcolumn* represents the number of the column which contains the kinases, in this case 9. The third argument represents optional proteome data (list of kinases expressed in HeLaS3, part of supplement) that can be used to filter the kinase list.

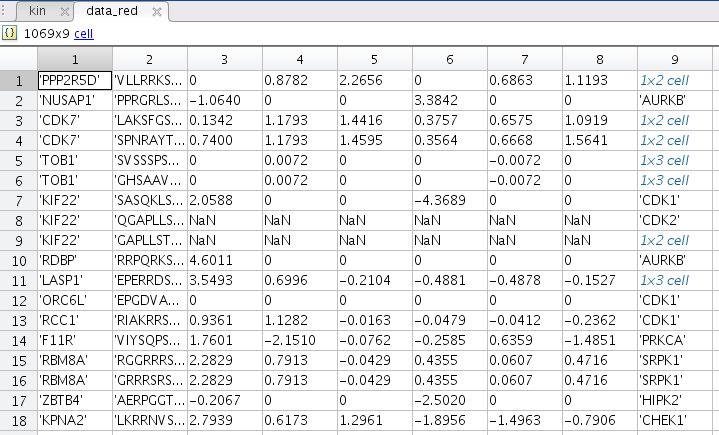


Fig. 4: Reduced dataset. Contains only those phosphosites for which a kinase is known from PSP.

A B

Fig. 5: List of kinases (kin) that are known to phosphorylate at least one phosphosite in the dataset (A). In this case, the list was filtered by proteome data (B).

**Step 1-3:**

[a]=CreateTT(data\_red,kcolumn,kin);

Creates a truth table (*a,* figure 6*)* with the dimensions phosphosite number x kinase number and 1s for true interactions. This will be used for activity estimation in the next step.

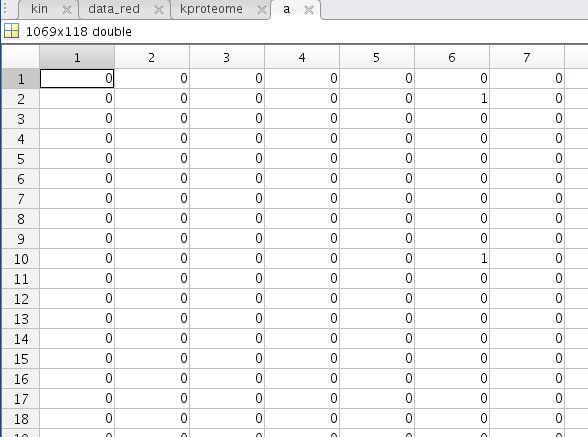


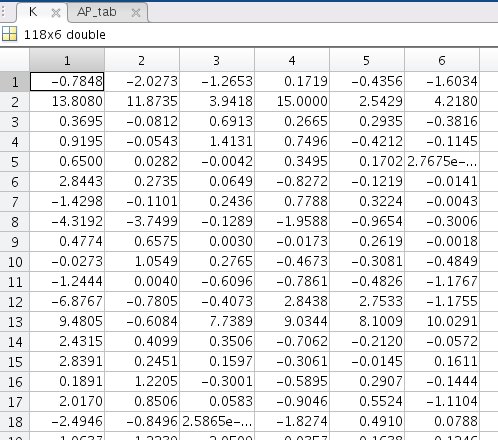
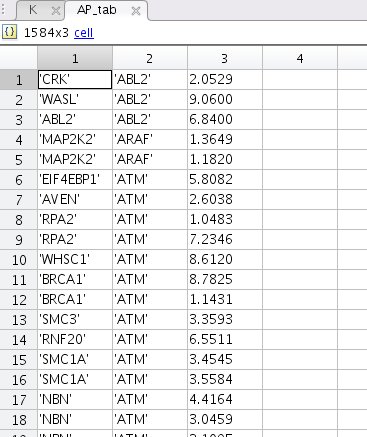
Fig. 6: Truth table. Contains a 1 if the interaction is true, 0 otherwise.

**Step 1-4:**

[AP,AP\_tab,K,cost,mincost]=FitActivities(a,data\_red,kin,ncon,iter);

FitActivities estimates the activities of the kinases in *kin* that best describe the measured data using the Matlab built-in function *fmincon*. *ncon* is the number of conditions in the dataset, this is 6 for the mitosis data. The number of iterations can be specified using the input argument *iter*. For reliable results this should be at least 5. Each iteration starts with randomized parameter vectors and cycles between an optimization of the affinity parameters by means of the global cost function and an optimization of the kinase activities for each condition by means of the local cost function. Since the gradients for the parameters are delivered by the cost functions, the ‘GradObj’ argument in the option settings should be turned to ‘on’. The while-loop continues as long as the difference between both costs is larger than a certain value, in this case 10. The upper and lower bounds for parameter estimation as well as the starting values should be modified according to the user´s needs. For mitosis data, we chose -15 and 15 for kinase activity bounds and randomized the starting values in this interval. The output variables *AP* and *K* contain the estimated affinity parameters and kinase activities, *cost* is a list of all calculated costs with *mincost* being the lowest. For a good fit, the value of *mincost* divided by the number of data points (number of phosphosites x number of conditions) should be smaller than 4. *AP\_tab* is a cell array assigning the affinity parameters to their respective interactions.

The function can also be executed on multiple cores using the parallel computing toolbox. In this case, the first for-loop needs to be replaced by a parfor-loop. Depending on your data size and hardware configuration the execution of this function can last a few hours and should thus be run over night. Figures 7 shows the output variables *K* and *AP\_tab.*

A B

Fig. 7: Estimated kinase activities that best describe the measured data. Each row stands for one of the 118 kinases, each column for one of the six conditions (A). Affinity parameters allocated to their respective interactions. Column 1 includes the targets, column 2 the kinases, column 3 the fitted affinity parameters (B).

Now that we have the kinase activities that best describe the data, we could stop the IKAP analysis at this point. The activities can be plotted, clustered, or used for pathway enrichment methods. However, we can also go on and examine the identifiability of the activities and affinities using step 1-5, or look for potential new kinase-target-links by means of IKAP part 2.

**Step 1-5:**

[plJ,plk]=IdentKin(ks,a,c,AP,K,data\_red,ncon);

This optional function explores if the fitted kinase activities are identifiable on the basis of the measured data, in other words, if they can be precisely determined. Since it would require exhaustive computation to test for all activities at all conditions, the user can choose for which kinases and conditions to test via the input arguments. *ks* is a vector containing the numbers of kinases that should be tested for as they appear in the list *kin*, *a* is the already established truth table, *c* the column number of the condition in *data\_red* the identifiability shall be calculated on, *AP* and *K* the variables containing the affinity parameters and kinase activities as they are produced by step 1-4, *data\_red* the reduced dataset and *ncon* the total number of conditions (as already used in the previous step). The output variable *plJ* (figure 9) contains the cost function values of the parameter space in the directions of the respective kinases. For each kinase, the functions tests seven different activity values. The other output variable *plk* (figure 10) contains the seven tested activities for each kinase. If a kinase is identifiable, a plot of its *plk* row (horizontal axis) against its plJ row (vertical axis) should display a clear minimum (figure 11). If the function is applied to many kinases it should be run over night.

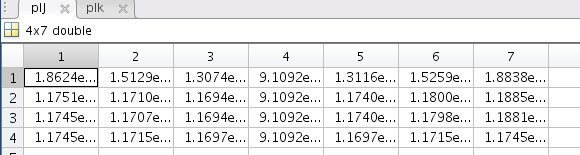


Fig. 8: Cost function values in the direction of four selected kinases.

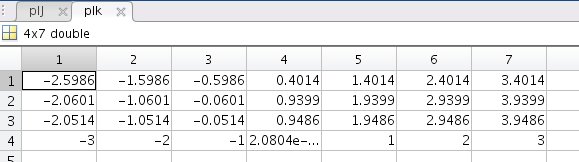


Fig. 9: Tested kinase activities.

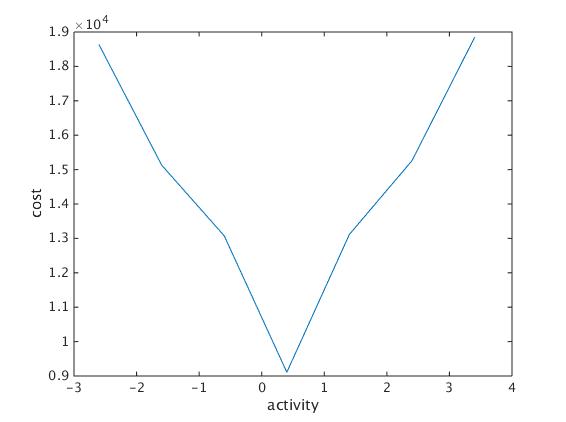


Fig. 10: Plot of plk(1,:) against plJ(1,:). The curve shows a clear minimum, thus the kinase is identifiable on the basis of the measured data.

**Step 2-1:**

[dist,valids]=ComputeDistances(data,ncon,kin,K);

In the second part of IKAP we use the estimated kinase activities to identify potential new kinase-target-links. This is done by correlating the kinase profiles to the measured phosphosite profiles. *data* is the complete data matrix (not the reduced one), *ncon* the number of conditions (6), *kin* the kinase list and *K* the fitted activities. The output is a matrix *dist* (figure 11) with phosphosites on the vertical and kinases on the horizontal axis. For each combination a p-value is given, deciphering the probability of the interaction. The other output *valids* contains the number of valid data points for each interaction.

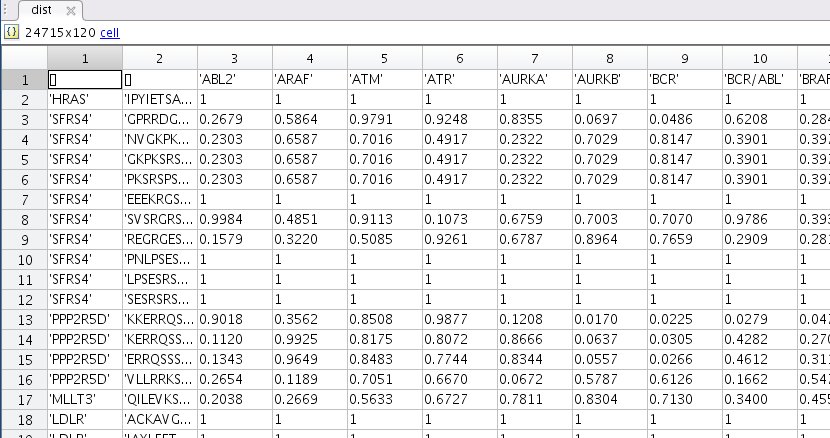


Fig. 11: Distance matrix containing a p-value for each kinase-target-interaction.

**Step 2-2:**

[psig,crit\_p]=MakePsig(dist,kin,q);

In this final step of part 2 we extract the significant kinase-target-links from the matrix *dist*. This is done by applying a user-defined false-discovery-rate q (0.05 in this case). The function makes use of the Benjamini-Hochberg method, the function fdr\_bh.m should thus be on the Matlab path. The output of this function is a cell array *psig* (figure 12) containing targets in column 1, kinases in column 2, motifs in column 3 and the respective p-value in column 4. It also outputs the critical p-value crit\_p, below which an interaction is regarded as being significant. These interactions can now be validated through IKAP part 3 or elsewise.

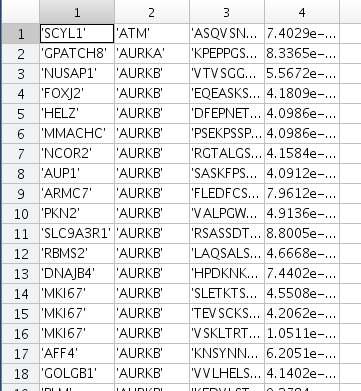


Fig. 12: Kinase-target-interaction with significant correlation coefficients.