JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

CallistoGen

modlab

ChemSpaceShuttle

for

Chemical Data Mining

Short manual

CallistoGen AG
Neuendorf 24b
D-16761 Hennigsdorf

&

Institute of Organic Chemistry and Chemical Biology
FB 14 Chemical and Pharmaceutical Sciences
modlab
Molecular Design Laboratory
Johann Wolfgang Goethe-University Frankfurt am Main
Germany

Alireza Givehchi

alireza.givehchi@chemie.uni-frankfurt.de

Contents

Initialize

Start

Start until distance level

Search empty classes

Save classification results

Save sorted classification results

Save classification results as html

Mail

Demo

Marked ID class and activity

View marked ID class (OFF/ON)

Save marked ID class

No activity 0

Activity 1

Activity 2

Activity 3

Activity 4

N dimensional data

Load data (TAB separated)

Set matrix mean to zero ON

Normalize matrix to [1,-1] ON

Select descriptor for X
(After loading the data, here will be viewed all the columns number)

Select descriptor for Y
(After loading the data, here will be viewed all the columns number)

Select descriptor for Z
(After loading the data, here will be viewed all the columns number)

Find 3 descriptors with the lowest cross-correlation

Non-linear transformation N to 3 dim
References non-linear transformation and evolutionary strategy

Non-linear transformation N to 3 dim (until adjusted STRESS)

Calculate STRESS
Reference for STRESS

Calculate STRESS ON/OFF

Keep the DissMat of orig. data ON/OFF

Non-linear Iterative Partial Least Squares NIPALS

References for NIPLAS

Save selected descriptors

Help

Result file types

Keyboard functions

File format

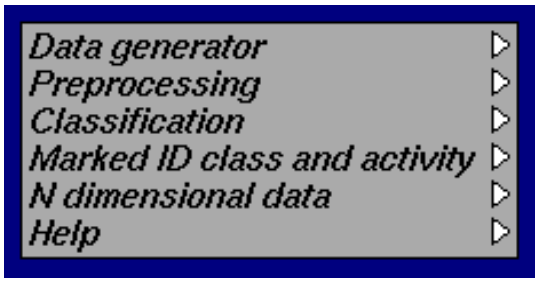
Input file

Output files

*.mid
*.n23
*.msc
*.sde
*.prp
*.kou
*.knet
*.res
*.cn.res

Viewing the menu window

If you hold down the right mouse button on the plot area of the ChemSpaceShuttle window the menu window will be viewed.



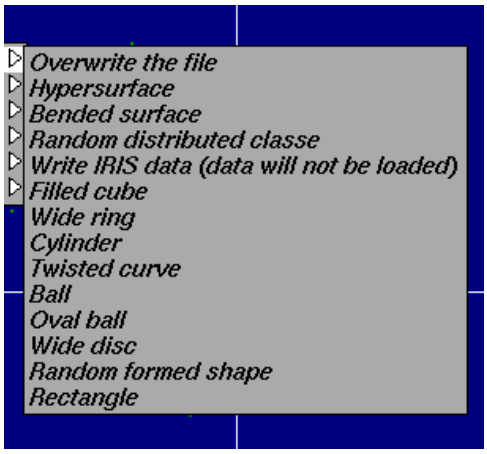
Keep the button down and move the mouse pointer on each item. If the item have a sub menu then the sub menu will be viewed. If the item has no sub menu and you release the mouse button then the item will be selected.

above

Data generator

This menu includes different items for creation of test data. The number of the created data are the same as the number of the input file vectors. For example, if you have a input file with 100 rows (each row includes the id, vector, and activity) in it and you select the submenu “Hypersurface” , then a data set will be created which distribution is like a Hypersurface. Take care that your original data will be overwritten if you selected the item “Overwrite the file”.

The creation of test data in e.g. combination with the functions in the menu “Classification” could help to understand the “Classification” functions



above

Overwrite the file

This item switches the overwriting of the input file on and off

Hypersurface

Creates test data which build a hyperspace. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Bended surface

Creates test data which build a bended surface. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Random distributed classes

Creates test data which build some data cluster with random distributed class centres and random distributed data points. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Write IRIS data (data will not be loaded)

This will create iris data and write it to the file ccss_iris.dat. The data will not be loaded and will not viewed. In order to load and view the data exit the program and start it with “ccss ccss_iris.dat”.

The iris test data could be created immediately if the program will be started with the parameter “irisdat”, i.d. “ccss irisdat”. The data set will created and the program exit. The iris data and the refernces are copied from the program “R”

We included tow new columns to the data set. Fisrt column is the ID column and the second column is last column is the activity column.

The first 6 rows of the file will be,

ID	Vector				Activity
3-setosa	4.9	3.0	1.4	0.2	3.0
4-setosa	4.7	3.2	1.3	0.2	3.0
5-setosa	4.6	3.1	1.5	0.2	3.0
6-setosa	5.0	3.6	1.4	0.2	3.0
7-setosa	5.4	3.9	1.7	0.4	3.0
8-setosa	4.6	3.4	1.4	0.3	3.0

After writing the ccss_iris file if the program will started with e.g.:

“ccss ccss_iris.dat setosa 3”

and transform the data set to 3-dimensional space then the data points with id setosa will be highlighted and the dot size will be set to 3. In this way one can see if the transformation to 3-dimensional space and the descriptors which are used are capable or not. If it will be started with e.g.:

“ccss ccss_iris.dat actmarked 4”

then all 3 classes will be highlighted in different color (see activity 4 and activity 1 for actmarked parameter).

Reference for IRIS data:

Edgar (1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2-5.
Internet home page the program R: http://www.r-project.org/

Filled cube

Creates test data which build a filled cube. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved.

Wide ring

Creates test data which build a wide ring. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Cylinder

Creates test data which build a cylinder. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Twisted curve

Creates test data which build a twisted curve. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Ball

Creates test data which build a ball. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Oval ball

Creates test data which build a oval ball. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Wide disc

Creates test data which build a wide disc. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

Random formed shape

Creates test data which build random distributed data set. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

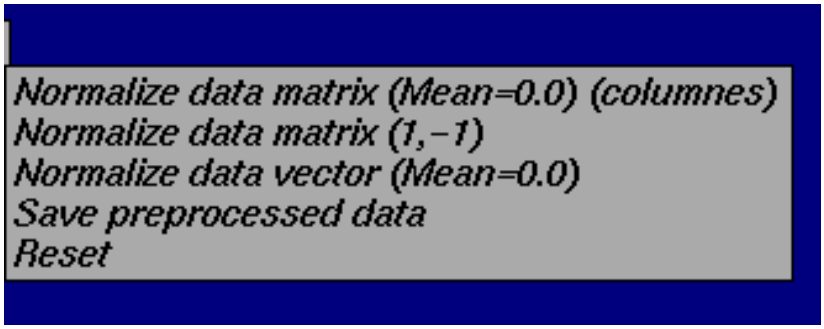
Rectangle

Creates test data which build a rectangle. The number of data points will be the equal to the number of the vector (number of rows) in the input file. The result could be preprocessed and saved

above

Preprocessing

This menu includes functions for normalization of the data. If you load a N-dimensional data set and transform it to 3-dimemsional space the normalization procedures will manipulate the transformed data and not the original N-dimensional data



Normalize data matrix (Mean=0) (column)

Normalization of all column will occur (zero mean or mean-centring) if this item will be selected. If the mean value of the whole data set have a distance from the zero point and you want to move the centre of the data set to the coordinate position x=0, y=0, and z=0 then select this menu item.

$$x = \frac{x}{\bar{x}} \; ; \quad y = \frac{y}{\bar{y}} \; ; \quad z = \frac{z}{\bar{z}} \; ;$$

$\bar{x}, \bar{y}, \bar{z}$ are mean values of each component x, y, and z

Normalize data matrix (1,-1)

Normalization of whole data set will occur if this item will be selected.

$$x = \frac{x}{(x_{\max} - x_{\min})} \; ; \; y = \frac{y}{(y_{\max} - y_{\min})} \; ; \; z = \frac{z}{(z_{\max} - z_{\min})}$$

Normalize data vector (Mean=0)

The normalization with this menu item make the vectors mean value equal to zero. This normalization is useful if you don't want to have a translation of the vectors.

$$x = x - mean; \quad z = z - mean; \quad z = z - mean; \quad mean = \frac{x + y + z}{3}$$

Save preprocessed data

The preprocessed data will be not saved until this item will selected. First normalize the data and than save the normalized data. There will be created a file with file type "*.prp". For example if the input file name is "test.dat" the result file name will be "test.dat.prp".

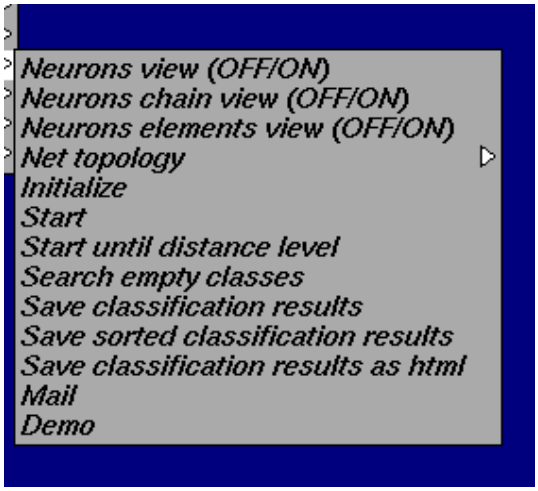
Reset

If you want to undo the preprocessing then select this item.

above

Classification

In this menu all the function for self-organizing neural net SOM are included. The function will sink in the data which are viewed in the 3-dimensional plot. If you e.g. load a N-dimensional data set and transform it to 3d space, the function of this menu will not sink on the original N-dimensional data set and the input vectors of the net will not be the N-dimensional data.



Neurons view (OFF/ON)

Visualization of the neuron position. Each neuron has a neuron weight vector. The vector is represented as a point in the s-dimensional space. Enlarge the points with “V”-key and downsize it with the “v”-key. This will sink only on the visualization of the points.

Neurons chain view (OFF/ON)

Visualization of the chain between the neuron. Remember the neuron are positioned on a map (1d, 2d, 3d). The map is not the same as the position of the neuron in the space.

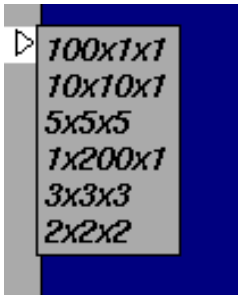
Neuron elements view (OFF/ON)

Visualization of the neuron members. Each neuron could be seen as a class. Each data point belong to on of the neuron depended on the distance of the data vector to the neuron weight vector. Through this menu item a line will be plotted between the data points and the neuron (class centre) to which they belong. So you could see if

the net is trained well or not. At the beginning of the training the lines are longer because the neuron are randomly distributed in the space.

Net topology

Sub menu for selection of the net topology. For a 1-dimensional map select e.g.10x1x1 or 1x200x1. For a 2-dimensional map select 10x10x1. It is possible to select 3-dimensional map (cube). For a 3-dimensional map select one of the items 5x5x5, 2x2x2, or 3x3x3.



100x1x1

This Creates an 1-dimensional map with 100 neuron.

10x10x1

This Creates a 2-dimensional map with 10 neuron in x and 10 neuron in 10 y direction.

5x5x5

This Creates a 3-dimensional map with 5 neuron in x, 5 neuron in y, and 5 neuron in z direction.

1x200x1

This Creates a 1-dimensional map with 200 neuron in y direction.

3x3x3

This Creates a 3-dimensional map with 3 neuron in x, 3 neuron in y, and 3 neuron in z direction.

2x2x2

This Creates a 3-dimensional map 3 neuron in x, 3 neuron in y, and 3 neuron in z direction.

Initialize

The neuron weight vectors will be initialized to random number if this item will be selected.

Keyboard button: F10

Start

Starts the training of the self-organizing net. The training of the net will be stopped if the max training step is reached. The max training step can be changed with the “k” and “K” button.

If you want to see how the neuron changes their position during the learning and how the map will be adapted to the data set, e.g. for demo purposes, then run the learning procedure from the menu item “Start until distance level”.

If it is desired to stop the training after a mean distance level is reached then run the learning procedure from the menu item “Start until distance level”.

The following algorithm is used to train the net:

1. Initialize the weights $\overline{W}$ to random number [-1,1] and initializing $\sigma(t)$ and $\varepsilon(t)$

2. Choose randomly a pattern vector $\vec{x}$ from the training data set which is represented in $\overline{X}$

3. Find the neuron s which weight vector $\overline{W}_{\vec{s}}$ has the smallest distance to the pattern vector

$$\left\|\vec{w}_{\vec{s}} - \vec{x}\right\| \le \left\|\vec{w}_{\vec{r}} - \vec{x}\right\| \qquad \forall \vec{r} \neq \vec{s} \qquad \vec{r}, \vec{s} \in PattDataSet$$

4. Define this neuron as the winner neuron and its position as the center of the stimulus

5. Update the weight matrix $\overline{W}$ which includes all the weight vectors

$$\vec{w}_{\vec{r}}^{new} = \vec{w}_{\vec{r}}^{old} + \varepsilon \times h_{\vec{r}\vec{s}} \times \left(\vec{x} - \vec{w}_{\vec{r}}^{old}\right) \qquad h_{\vec{r}\vec{s}} = e^{-\left(\frac{\left|\vec{r}-\vec{s}\right|^2}{2\sigma^2}\right)}$$

6. Update the value of $\sigma(t)$ and $\varepsilon(t)$ and repeat the procedure from the second point until the max. learning step $t_{\max}$ is not arrived.

$$\varepsilon(t) = \varepsilon_i \times \left(\frac{\varepsilon_f}{\varepsilon_i}\right)^{\frac{t}{t_{\max}}} \qquad \sigma(t) = \sigma_i \times \left(\frac{\sigma_f}{\sigma_i}\right)^{\frac{t}{t_{\max}}}$$

ε_i initial value of ε ; ε_f end value of ε ; σ_i initial value of σ ; σ_f end value of σ

Keyboard button: F11

Start until distance level

Starts the training of the self-organizing net. The training will be stopped either if max training step is reached (the max training step can be changed with the “k” and “K” button) or if the mean distance value between the neuron weights and the data vectors reaches the initiated mean distance value. The mean distance level could be changed with the keyboard button “+” and “-“. If the max training step is reached and the desired mean distance level is not reached the procedure will be stopped too.

If you want to see how the neuron changes their position during the learning and how the map will be adapted to the data set, e.g. for demo purposes, then run the learning from this menu item.

Search empty classes

After the training of the self-organizing net it could be that some neuron (classes) have no member. This item print the neuron index which are empty. This could help to choose the right map size or to check if the net is trained enough or not.

Save classification results

After the training of the self-organizing net the classification procedure will start automatically. This item saves the result of the classification. The neuron topology information and neuron weights will be save too. There will be created different result files. The file names begin with the input file name plus new file type. For example if the input file name is “test.dat” then one of the files will have the file name “test.dat.knet”.

Below the file types and their purpose are written,

knet	Self-organizing net parameter and trained weights

kou	Self-organizing net classification result
kou.res	relative number of active compounds in each neuron
kou.cn.res	number of active compounds in each neuron

Save sorted classification results

Not implemented!

Save classification results as html

Only the self-organizing net classification result (file type “kou”) will be save in html format.

Mail

The self-organizing net classification result (file type “kou”) will be send to your user e-mail address (for later version it is intended to make it possible to send the file to a selected e-mail address).

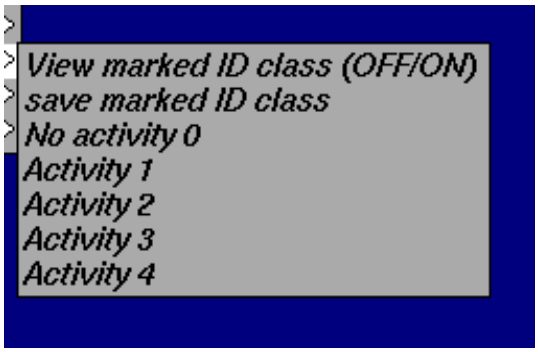
Demo

Starts a demo for the SOM training and view how a map will be adapted to the data set.

above

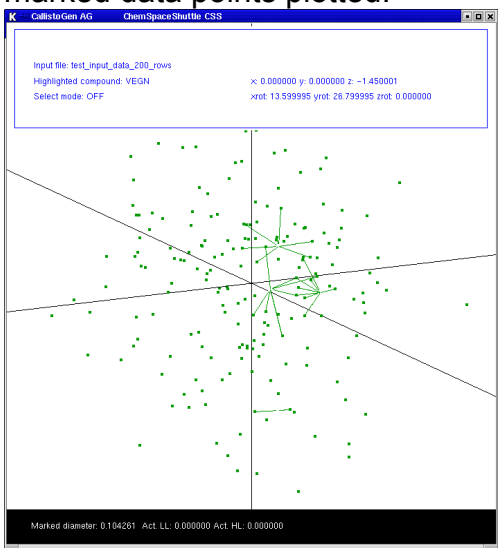
Marked ID class and activity

This menu item includes items for viewing the different compound classes after their activity and for highlighting compounds with marked id.



View marked ID class (OFF/ON)

This switches [on/off] viewing the data points which are in the environment of the marked id's. A line between the marked data point and the compound belongs to it will be plotted. If no data point are in the environment (in the adjusted radius) of the marked then no lines will plotted. To adjust the environment radius push the button "G" to enlarge and "g" to downsize. In the figure below are environment of the marked data points plotted.



To highlight the data points with certain ID start ChemSpaceShuttle with

"ccss filename id dotSize".

E.g. if the input file name is "test.dat" and you want to highlight all data points with the string "cox" in their id's and you want to plot them with dot size 2 then start the program with

"ccss test.dat cox 2"

Now all the data points with the sub-string "cox" in their id will be plotted with different color as the data without the sub-string "cox".

The position of the marked data points could be seen as the middle point of classes and the data points in their environment (with adjustable radius) as the member of the classes.

The data points size which id are marked could be changed with key "B" and "b".

Save marked ID class

Saves the marked data points with the data points laying in their environment (file type *.mid)

See menu item "View marked ID class (OFF/ON)"

No activity 0

Switches off highlighting after the activity value (see also sub-menu "Activity 1")

Activity 1

Selecting this menu item will highlight data pointes after the activity value in the following order.

```
<   0.0
== 0.0
>   0.0          <= 1.0
>   1.0      && <= 10.0
>   10.0     && <= 100.0
>   100.0
```

To highlight data points after the activity value of the compounds (see also input file format) it is also possible to start the program with

```
ccss filename actmarked modNr
```

E.g. if your input file name is test.dat and you want have the mod 1 (see sub-menu Activity 2 to 4) then start the program with

```
ccss test.dat actmarked 1
```

Activity 2

Selecting this menu item will highlight data pointes after the activity value in the following order (see also sub-menu "Activity 1"):

```
>= 0.0 && <= 1.0
else
```

This has the same effect as if the program would be started with

```
ccss filename actmarked 2
```

Activity 3

Selecting this menu item will highlight data pointes after the activity value in the following order (see also sub-menu "Activity 1"):

```
<= 0.5
>   0.5
else
```

This has the same effect as if the program would be started with

ccss filename actmarked 3

Activity 4

Selecting this menu item will highlight data pointes after the activity value in the following order (see also sub-menu “Activity 1”):

```
<    -1.0
>=   -1.0  &&  <=  0.0
>     0.0  &&  <    1.0
>=    1.0  &&  <    2.0
>=    2.0       <    3.0
>=    3.0       <    4.0
>=    4.0
```

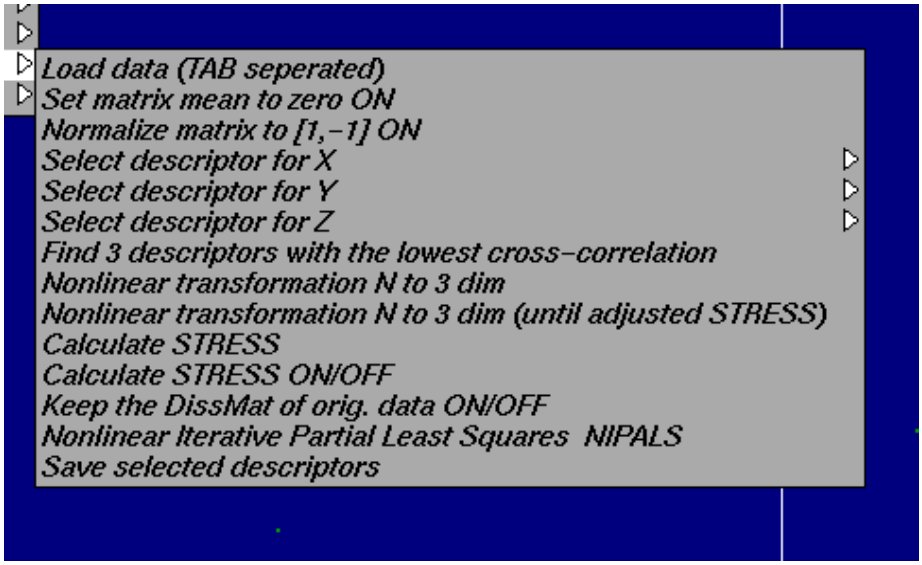
This has the same effect as if the program would be started with

ccss filename actmarked 4

above

N dimensional data

This menu includes mainly sub-menus for Transformation of more than 3 dimensional input data set to 3 dimensional space. The most important transformation is behind the items “Non-linear transformation N to 3 dim” and “Non-linear transformation N to 3 dim (until adjusted STRESS)”



Load data (TAB separated)

This item must be selected before transforming the N dimensional data to 3 dimension.

Set matrix mean to zero ON

If the data matrix should be normalized (column: mean to zero) each time after loading and manipulating them then this item should be switched ON.

$$x = \frac{x}{\bar{x}} \; ; \quad y = \frac{y}{\bar{y}} \; ; \quad z = \frac{z}{\bar{z}} \; ;$$

$\bar{x}, \bar{y}, \bar{z}$ are mean values of each component x, y, and z

This function can be selected manually from the menu "Preprocessing"

Normalize matrix to [1,-1] ON

In order to normalize the matrix each time after manipulating the data set, this item must be switched on

$$x = \frac{x}{(x_{\max} - x_{\min})} \; ; \; y = \frac{y}{(y_{\max} - y_{\min})} \; ; \; z = \frac{z}{(z_{\max} - z_{\min})}$$

This function can be selected manually from the menu "Preprocessing"

Select descriptor for X

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

The data set which is loaded consists of rows (input vectors) and columns (variables or input vector components). This menu item make it possible to select a variable or column to be as the x values.

(After loading the data, here will be viewed all the columns number)

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

Select descriptor for Y

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

The data set which is loaded consists of rows (input vectors) and columns (variables or input vector components). This menu item make it possible to select a variable or column to be as the y values.

(After loading the data, here will be viewed all the columns number)

The menu item „Load data (TAB separated)" must be selected one time before selecting this item

Select descriptor for Z

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

The data set which is loaded consists of rows (input vectors) and columns (variables or input vector components). This menu item make it possible to select a variable or column to be as the z values.

(After loading the data, here will be viewed all the columns number)

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

Find 3 descriptors with the lowest cross-correlation

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

If this menu item will be selected then the three columns of the data set with the lowest cross-correlation will be set to x, y, and z.

Non-linear transformation N to 3 dim

The menu item „Load data (TAB separated)" must be selected one time before selecting this item

In order to transform the loaded N-dimensional data set to a 3-dimensional space this menu item must be selected. The transformation will occur through a multiplayer neural net (as a non-linear encoder networks) which will be trained with evolutionary strategy. The menu item "Non-linear transformation N to 3 dim (until adjusted STRESS)" makes such a transformation too but with a additionally function.

References non-linear transformation and evolutionary strategy

Schneider, G., Wrede, P., J. Mol. Evol., 1993, 36, 586-595.
Schneider, G., Wrede, P., Prog. Biophys. Mol. Biol. 1998, 70, 175-222.
Schneider, G., So, S., Adaptive Systems in Drug Design, Biotechnology Intelligence Unit 5, Eureka, 2002.

Non-linear transformation N to 3 dim (until adjusted STRESS)

The menu item „Load data (TAB separated)" must be selected one time before selecting this item

In order to transform the loaded N-dimensional data set to a 3-dimensional space this menu item must be selected. The transformation will occur through a multiplayer neural net (as a non-linear encoder networks) which will be trained with evolutionary strategy. The calculation will be repeated until a desired STRESS value is reached. The menu item "Non-linear transformation N to 3 dim" makes also such a transformation but without the calculation of STRESS.

Calculate STRESS

The menu item „Load data (TAB separated)" must be selected one time before selecting this item.

The selection of this menu item will calculate the STRESS value. STRESS is the Kruskal's Standardized Residual Sum of Squares.

$$S = \sqrt{\frac{\sum_i \sum_j \left(d_{ij}^t - d_{ij}^o\right)^2}{\sum_i \sum_j d_{ij}^{o\,2}}}$$

d_{ij}^t is the Euclidian distance between the 3D vectors i and j (t for transformation), and d_{ij}^o is the Euclidian distance between the original descriptor vectors i and j (o for original).

If you didn't select the menu item "Calculate STRESS ON/OFF" in order to calculate STRESS automatically after each time the transformation will be executed then you can select this menu item in order to calculate the STRESS value manually.

Reference for STRESS

Kruskal, J.B., Nonmetric multidimensional scaling: a numerical method, Psychometrika, 1964, 29, 115-129.

Calculate STRESS ON/OFF

If it is set to ON calculates the STRESS value automatically after each transformation, if not then no calculation of STRESS will be executed (see menu item "Calculate STRESS").

Keep the DissMat of orig. data ON/OFF

If it is set to ON then the calculated dissimilarity matrix of the original N-dimensional data set will be kept for following transformation. This will save calculation time.

Non-linear Iterative Partial Least Squares NIPALS

The menu item „Load data (TAB separated)" must be selected one time before selecting this item

This menu item executes the non-linear iterative partial least squares (NIPALS) for principle component analysis.

The algorithm which is used for NIPALS is declared below:

The data set will be normalized (mean-centring and unit variance scaling)

Step 1: Set the array of the $\vec{s}_i$ to the first column $\vec{x}_1$ of the input vector matrix $\overline{X}$ (each column of this matrix represents different descriptors and each row represents a different compound).

Step 2: Calculation of $\vec{e}_i$: $\vec{e}_i^{\,T} = \vec{s}_i^{\,step2\,T}\,\overline{X}$; where T means transpose.

Step 3: Normalization of the vector $\vec{e}_i$ to the length 1: $\vec{e}_i = \frac{\vec{e}_i}{\|\vec{e}_i\|}$.

Step 4: Calculation of $\vec{s}_i^{\,step4} = \overline{X} \cdot \frac{\vec{e}_i}{\|\vec{e}_i\|^2}$.

Step 5: Compare $\vec{s}_i^{\,step4}$ from step 4 to from step 2; if the sum of squared residual is smaller than 1.0E-10 then go to step 6 else set $\vec{s}_i^{\,step2} = \vec{s}_i^{\,step4}$ and go to step 2 and calculate $\vec{e}_i$ again.

Step 6: Calculate the residual $\overline{R} : \overline{R} = \overline{X} - \vec{s}_i \vec{e}_i^{\,T}$ and set $\overline{X} = \overline{R}$. If $i = p$ or dot-product of R < 1.0E-10 then stop the calculation, else go to step 2.

References for NIPLAS

Wold H., Nonlinear estimation by iterative least squares procedures. in Research papers in statistics (ed. David F.), Wiley & Sons, New York. 1966, 441-444
Wold, H. Estimation of principal components and related models by iterative least squares. in Multivariate analysis (ed. Krishnaiah P. R.) Academic Press, New York, 1966, 391-420.
Wold, S., Cross-validatory estimation of the number of components in factor and principal components models. Technometrics, 1978, 20: 397-405.
Eriksson, L., Johansson, E., Kettaneh-Wold, N., Wold, S., Multi- and Megavariante Data Analysis, Principles and Applications
Gustafsson, m.G., A probabilistic Derivation of the Partial Least-Squares Algorithm, JCICS, 2001, 41, 288-294.

Save selected descriptors

The menu item „Load data (TAB separated)" must be selected one time before selecting this item

After the transformation of the data set to a 3-domensioanl space the results can be saved through this menu item. The type of the output file will be "sde"

above

Help

Prints the list of helps about the keyboard functions.

above

Result file types

mid	Marked id classes
n23	Result of N-D to 3-D transformation
msc	Manually selected classes
sde	Selected descriptor vectors (from N to 3 dim)

prp	Preprocessed data
knet	Self-organizing net parameter and trained weights
kou	Self-organizing net classification result
kou.res	Relative number of active compounds in each neuron
kou.cn.res	Number of active compounds in each neuron

above

Keyboard functions

h,H	Shows this help
T	zoom in (slow)
T	zoom in (fast)
Z	zoom out (slow)
Z	zoom out (fast)
w	move upward
s	move downward
d	move right
a	move left
▲	rotating the coordinates upward arrow button (arrow button)
▼	rotating the coordinates downward (arrow button)
►	rotating the coordinates right (arrow button)
◄	rotating the coordinates left (arrow button)
M	enlarge data points
m	downsize data points
b	downsize the selected data points
B	enlarge the selected data points
y	downsize the selection radius (only in selection mode F6 see also F9)
Y	enlarge the selection radius (only in selection mode F6 see also F9)
l	Multiple selection mode (only in selection mode F6 see also F9)
p	print mode print the selected data (only in selection mode F6, see also F9)
v	downsize the neuron points (for visualization)
V	enlarge the neuron points (for visualization)
k	Decrease max learning step of SOM
K	increase max learning step of SOM
+	Decrease the max value of mean SOM distance (stop the training if mean SOM distance is reached
-	increase the max value of mean SOM distance (stop the training if mean SOM distance is reached
g	downsize the radius of the selected id environment
G	enlarge the radius of the selected id environment
f	Decrease the max activity value data set with a activity higher than this value will be not viewed
F	decrease the max activity value data set with a activity higher than this value will be not viewed
r	decrease the min activity value data set with a activity lower than this value will be

	not viewed
R	increase the min activity value data set with a activity lower than this value will be not viewed
c	change the background color
x	cube [on/off]
X	coordinate axes [on/off]
>	Manually selection of descriptors (upward)
<	Manually selection of descriptors (downward)
.	decrease the max STRESS value
Cntr .	increase the max STRESS value
F1	window size
F2	increase the rotating tempo in all direction
F3	decrease the rotating tempo in all direction
F4	stop rotating
F5	reset the view to the start position
F6	selection mode [on/off]
F7	save the selected data
F8	Info window (top window) [on/off]
F9	take the selected data (if in multiple selection mode)
F10	Initialisation of the neuron positions
F11	classification (also selectable from the menu)
F12	save the classification results

File format

Input file

ID	Input vector				Activity
YRSGLMCMV	0.0000	0.0141	0.0141	0.0141	1.0
YRASIIAVV	0.0000	0.0143	0.0143	0.0143	1.0
FRSGIIAVV	0.0000	0.0147	0.0147	0.0147	1.0

ID could be any string with the length of 5000 (for example SMILES). Activity must be a floating value. You could use this value for example for classification of active peptides. Between ID and activity are the vector components (floating value). The input vector dimension could be >=3. Note: The column separator must be TAB.

Output files

***.mid**

This file contains the compounds which are in the dedicated distance to the marked compounds.

File format:

ID	Input vector	Activity
YRSGLMCMV	0.0000 0.0141 0.0141	1.0
YRASIIAVV	0.0143 0.0143 0.0143	1.0
FRSGIIAVV	0.0147 0.0147 0.0147	1.0

The same format as the input file has. The only different is the dimension of the input vector, which is always 3.

***.n23**

This file contains the results of the ND to 3D non-linear transformation

File format:

ID	Input vector	Activity
YRSGLMCMV	0.0000 0.0141 0.0141 0.0141	1.0
YRASIIAVV	0.0000 0.0143 0.0143 0.0143	1.0
FRSGIIAVV	0.0000 0.0147 0.0147 0.0147	1.0

The same format as the input file has. The only different is the dimension of the input vector, which is always 3.

***.msc**

This file contains the manually selected compounds.

File format:

ID	Input vector	Activity
YRSGLMCMV	0.0000 0.0141 0.0141 0.0141	1.0
YRASIIAVV	0.0000 0.0143 0.0143 0.0143	1.0
FRSGIIAVV	0.0000 0.0147 0.0147 0.0147	1.0

***.sde**

This file contains the compounds and their manually selected descriptor vectors (from N to 3 dim).

File format:

ID	Input vector				Activity
YRSGLMCMV	0.0000	0.0141	0.0141	0.0141	1.0
YRASIIAVV	0.0000	0.0143	0.0143	0.0143	1.0
FRSGIIAVV	0.0000	0.0147	0.0147	0.0147	1.0

***.prp**

This file contains the compounds and their preprocessed descriptor vectors.

File format:

ID	Input vector				Activity
YRSGLMCMV	0.0000	0.0141	0.0141	0.0141	1.0
YRASIIAVV	0.0000	0.0143	0.0143	0.0143	1.0
FRSGIIAVV	0.0000	0.0147	0.0147	0.0147	1.0

***.kou**

```
# ParFile: test.par
# TrainPattFile: test.in
# NetFile: test.net
# fErrorLevel: 100.000000
# Y-index X-index  Z-index  Dist     Seq  active/inactive
1    1    1        0.393329        YRSGLMCMV       1.000000
1    1    1        0.432605        YRASIIAVV       1.000000
1    1    1        0.447198        FRSGIIAVV       0.000000
```

^	^	^	^	^	^
|	|	|	|	|	|
Y	X	Z	Distance	MolName	activity

***.knet**

The weight vector of neuron of each row of the map will be printed in a separate line. For example the first line includes all the weights of the neuron in the first line on the map. The last lines signed with # at the beginning of the line include information about the parameter of the net and information about the used file.

Example for the last lines in the net file

```
# NetNr 1
# NetInputNr 150 NetEpsI 0.900000 NetSigI 0.300000 NetEpsF 0.010000 NetSigF 0.010000
  NetXNeuronNr 10 NetYNeuronNr 10 NetLernStep 2 Normalize 1
```

TrainPattFile test.in ParFile test.par

***.res**

Example file:

```
11.76 00.00 00.00 00.00 00.00
 63.41 00.00 00.00 00.00 00.00
 57.77 62.85 00.00 00.00 37.14
 32.33 00.00 32.25 00.00 00.00
 63.41 00.00 00.00 00.00 00.00
 57.77 62.85 00.00 00.00 37.14
 11.76 00.00 00.00 00.00 00.00
 # NetXNeuronNr 5  NetYNeuronNr 7
```

The floating values show the relation of the number of the active compound to the number of all compounds in each neuron. In the last row the size (5 neuron in the x direction and 5 neuron in the y direction) of the map is printed. This file could be used for the classification of active and non-active peptides. If you want to print color map of this file execute showcolorkores. The right activity must be set in the input file (see komap -ih) if this file will be used!

***.cn.res**

Output file (*.cn.res) for relation of the number of the compound in the neuron to the number of compounds in all neuron.

Example file:

```
 11.76 00.00 00.00 00.00 00.00
 63.41 00.00 00.00 00.00 00.00
 57.77 62.85 00.00 00.00 37.14
 32.33 00.00 32.25 00.00 00.00
 63.41 00.00 00.00 00.00 00.00
 57.77 62.85 00.00 00.00 37.14
 11.76 00.00 00.00 00.00 00.00
 # NetXNeuronNr 5  NetYNeuronNr 7
```

The floating values show the relation of the number of the compound in the neuron to the number of compounds in all neuron. In the last row the size (5 neuron in the x direction and 5 neuron in the y direction) of the map is printed. This file could be used to see the neuron occupation. If you want to print color map of this file execute showcolorkore.