emergent reference

Keyboard shortcuts

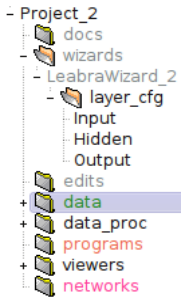
Global project

Ctrl+s	Save project.
Ctrl+left	Backwards in navigation history
Ctrl+right	Forwards in navigation history
F5	Refresh.
Tab	Forward through interface
Shift+Tab	Backwards through interface

Tree browser and program code

Any 1-3 chars	Find as you type
Alt+f	Find from selected node.
Ctrl+i	New item below cursor.
Ctrl+f	Expand this node.
Shift++	Expand this node.
Ctrl+b	Collapse this node.
-	Collapse this node.
Ctrl+spacebar	Selection mode.
Ctrl+p	(select) Previous element.
Ctrl+n	(select) Next element.
Ctrl+d	Delete selected item(s).
Delete	Delete selected item(s).
Ctrl+c	Copy selected element(s).
Ctrl+x	Cut selected element(s).
Ctrl+v	Paste element(s).
Ctrl+u	Page up.
Ctrl+v	Page down.
Ctrl+g	Deselect.
Esc	Deselect.

New elements in left tree browser



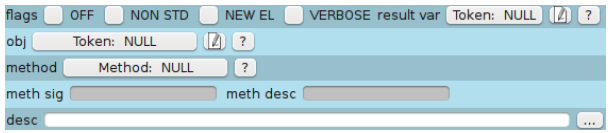
do Ctrl+i	New Doc
da Ctrl+i	New DataTable
la Ctrl+i	New Layer
P Ctrl+i	New Project
pr Ctrl+i	New Program
n Ctrl+i	New Network
sp Ctrl+i	New Spec

New elements in program code

These sequences insert new items and then take you back.

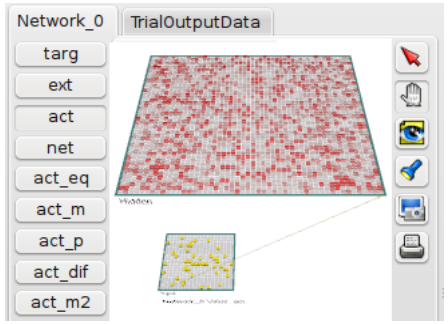
obj Ctrl+i Type Ctrl+left,left	New obj of Type
var Ctrl+i Ctrl+left,left	New var
arg Ctrl+i Ctrl+left,left	New arg
fun Ctrl+i Ctrl+left,left	New fun
init Ctrl+i Name Ctrl+left,left	New init code Name
prog Ctrl+i Name Ctrl+left,left	New prog code Name

Middle panel edit dialogs



Tab	Next element.
Shift+tab	previous element.
Up	(numeric field) Increase value.
Down	(numeric field) Decrease value.
Up	(dropdown) Move up.
Down	(dropdown) Move down.
ESC	Revert changes.
Ctrl+Enter	Apply changes.
Spacebar	(buttons) Open token chooser.
Spacebar	(flags) Check/uncheck flag.
Ctrl+l	(expression fields) Lookup information.

3D network and graph viewer



i	Interact (mouse cursor).
v	Camera view (hand).
a	View all (eyeball) (broken).
s	Seek (flashlight) (broken).
Shift+mouse	Drag in x,y plane.
Middle mouse scroll	Zoom in/out in z plane.

Programing

taDataProc::

Columns category

```
ConcatCols ( DataTable* dest, DataTable* src_a,...)
```
Concat two tables, preserving all data
```
Join(DataTable* dest,DataTable* src_a,DataTable* src_b,...)
```
Left, right and inner join two tables

Copy category

```
AppendRows(DataTable* dest,DataTable* src)
```
Append rows of src to dest
```
ConcatRows(DataTable* dest,DataTable* src_a, ...)
```
Concatenate rows from all src tables into dest
```
CopyCommonColData(DataTable* dest,DataTable* src)
```
Append data from src to dest for all common cols
```
CopyCommonColsRow(DataTable* dest,DataTable* src,
                  int dest_row, int src_row)
```
Append data from src to dest for all common cols
```
CopyData(DataTable* dest,DataTable* src)
```
Destructively copy data from src to dest
```
ReplicateRows(DataTable* dest,DataTable* src,int n_repl)
```
Destructively replicate rows of src into dest n_repl times

Order category

```
Group(DataTable* dest,DataTable* src,DataGroupSpec* spec)
```
Group data from src into dest according to spec
```
Permute(DataTable* dest, DataTable* src)
```
Randomly reorder the rows of src table into dest
```
Sort(DataTable* dest,DataTable* src,DataSortSpec* spec)
```
Sort src data into dest according to sort spec
```
SortInPlace(DataTable* dt,DataSortSpec* spec)
```
Sort data in place according to sort spec

Select category

```
SelectRows(DataTable* dest,DataTable* src,DataSelectSpec* spec)
```
Select rows of src into dest according to spec
```
SplitRows(DataTable* dest_a,DataTable* dest_b,...)
```
Split src rows that mach spec into dest_b, otherwise dest_a

taDataGen::

Basic category

```
Clear(DataTable* data,const taString& col_nm,float val=0.0)
```
Clear all data. Set all data to val if provided.
```
SimpleMath(DataTable* data,const taString& col_nm,...)
```
Apply simple math op to all vals in float_Matrix col

Distance category

```
LastMinDist(DataCol* da,int row,...)
```
returns min distance between nth pattern and all previous
```
LastMinMaxDist(DataCol* da,int row,float& max_dist,...)
```
Returns min and max distance between nth patte

Draw category

`RenderLine(float_Matrix* mat,int xs, int ys, int xe,...)`
Render a line from,to start,end
`RenderWideLine(float_Matrix* mat,int xs, int ys,...)`
Render a wide line from,to start,end
`WritePoint(float_Matrix* mat,int x,int y,...)`
Write a single point

Files category

`GetDirFiles(DataTable* dest,...)`
Read file names from given directory into rows of the data table

Lists category

`CombineFrequencies(DataTable* freq_output,...)`
Operate on input items,freqs into output freqs
`CrossLists(DataTable* crossed_output,...)`
Creates a full set of combination of elements from two or more lists.
`ProbSelectColNo(DataTable* data_table,...)`
Select a column number from data table based on probabilities associated with different columns.
`ProbSelectRow(DataTable* data_table,...)`
Randomly generate events based on a set of probabilitis for given options at each point.
`ReplicateByFrequency(DataTable* repl_output,...)`
Replicate input by the number in the frequency column times the total_number value.
`SampleByFrequency(DataTable* repl_output,...)`
Sample the items in the input data as a function of the probability value given in the frequency column, with n_samples taken per row .
`SortedPermutations(DataTable* dest, int n)`
Generate a sorted list of all possible n! permutations of the digits 1..n in sorted order and write them to destination data table dest.

Random category

`AddNoise(DataTable* data,...)`
Add random noise of specified type to the patterns.
`AddNoiseMat(float_Matrix* mat,...)`
Add random noise to given pattern.
`FlipBits(DataTable* data,...)`
Flip n_off bits from 1's to 0's, and n_on bits from 0's to 1's in float matrix column col_nm.
`FlipBitsMat(float_Matrix* mat,...)`
Flip n_off of the 1 bits into the 0 state, and n_on of the 0 bits to the 1 state. `PermutedBinary(DataTable* data,...)`
Create permuted binary patterns of n_on on_vals (1's) and rest off_vals (0's) in given col (must be float matrix).
`PermutedBinaryMat(float_Matrix* mat,...)`
Set matrix values to permuted binary pattern of n_on on_vals and rest off_vals.
`PermutedBinary_MinDist(DataTable* data,...)`
Create permuted binary patterns with dist minimum hamming distance (or dist max_correl).

SubMatrix category

`ReadToSubMatricies(DataTable* src,...)`
For making larger patterns out of smaller ones (sub-matricies) and vice-versa.
`WriteFmSubMatricies(DataTable* dest,...)`
For making larger patterns out of smaller ones (sub-matricies) and vice-versa.

taDataAnal::

Clean category

`SmoothExp(DataTable* smooth_data,...)`
Exponential smoothing: compute the exponentially-convolved average for all the numeric fields of source data, using an exponential kernel of given half-width and exponent.
`SmoothGauss(DataTable* smooth_data,...)`
Gaussian smoothing
`SmoothPow(DataTable* smooth_data,...)`
Power-function smoothing
`SmoothUniform(DataTable* smooth_data,...)`
Uniform smoothing
`TimeAvg(DataTable* time_avg_data,...)`
Compute the time average for all the numeric fields of source data, according to the given avg_dt.

Correlation category

`CorrelMatrix(float_Matrix* correl_mat,...)`
Compute correlation matrix across rows for given matrix data column in src_data datatable.

Distance category

`CrossDistMatrix(float_Matrix* dist_mat,...)`
Compute cross distance matrix between two different matrix data columns in src_data_a and src_data_b datatables.
`DistMatrix(float_Matrix* dist_mat,...)`
Compute distance matrix for given matrix data column in src_data datatable.

Graph

`Matrix3DGraph(DataTable* data,...)`
Prepare data for a 3D matrix graph, where data is plotted by X and Z axis values – sorts data by X then Z, then adds a duplicate copy of data sorted by Z then X, which produces a matrix grid in a graph view plot (turn off the Z neg draw flag).

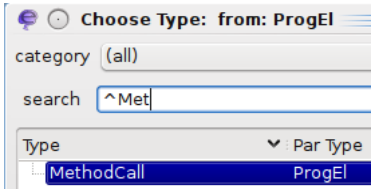
HighDim

`Cluster(DataTable* clust_data,...)`
Produce a hierarchical clustering of the distances between patterns in given data column from source data, with labels from given name_col_nm, using given distance metric.
`MDS2dPrjn(DataTable* prjn_data,...)`
Perform multidimensional scaling on the distance matrix (computed according to metric, norm, tol parameters) of patterns in column name across rows, putting the resulting projections into prjn_data.
`PCA2dPrjn(DataTable* prjn_data,...)`
Perform principal components analysis of the correlations of patterns in given columm across rows, plotting projections of patterns on the given principal components in the data table.
`PCAEigens(float_Matrix* eigen_vals,...)`
Get principal components analysis (PCA) eigenvalues and eigenvectors of correlation matrix across rows for given matrix column name in source data
`RowPat2dPrjn(DataTable* prjn_data,...)`
Project all rows according to their projection onto the two specified rows of patterns using given distance metrics.

Stats

`RegressLinear(DataTable* src_data,...)`
Compute linear regression (least squares fit of function y = mx + b) to given data.

Program code elements



Press Ctrl+i seq Enter as fast as you can, where seq is defined below as the shortest sequence needed to put that program element at the top of the chooser list. No need to wait for visual confirmation of the choice.

Ctrl

`ForLoop`	f.
`DoLoop`	do.
`WhileLoop`	w.
`If`	ife.
`IfCont`	ifc.
`IfBreak`	ifb.
`IfReturn`	ifr.
`CodeBlock`	co.
`UserScript`	u.

Var/Fun

`ProgVars`	progvars.
`AssignExpr`	as.
`VarIncr`	v.
`MemberAssign`	me.
`MethodCall`	met.
`MemberMethodCall`	me Tab Ctrl+n,n.
`FunctionCall`	fu Tab Ctrl+n.
`ReturnExpr`	ret.
`ProgramCall`	prog Tab Ctrl+n,n.
`ProgramCallVar`	prog Tab Ctrl+n,n,n.
`OtherProgramVar`	prog Tab Ctrl+n,n,n.

Print/Args

`PrintExpr`	p.
`PrintVar`	p Tab Ctrl+n.
`Comment`	com.
`StopStepPoint`	sto.
`ProgVarFmArg`	pro.
`MemberFmArg`	me Tab Ctrl+n.
`DataColsFmArgs`	dataco.
`RegisterArgs`	re.

Misc Fun

`StaticMethodCall`	st.
`MathCall`	m.
`RandomCall`	r.
`MiscCall`	mi.
`DataProcCall`	datap.
`DataAnalCall`	d.
`DataGenCall`	datag.
`ImageProcCall`	im.

Data

`DataLoop`	datal.
`ResetDataRows`	res.
`AddNewDataRow`	a.
`DoneWritingDataRow`	don.
`DataVarProg`	datav.
`DataVarProgMatrix`	datav Tab Ctrl+n.