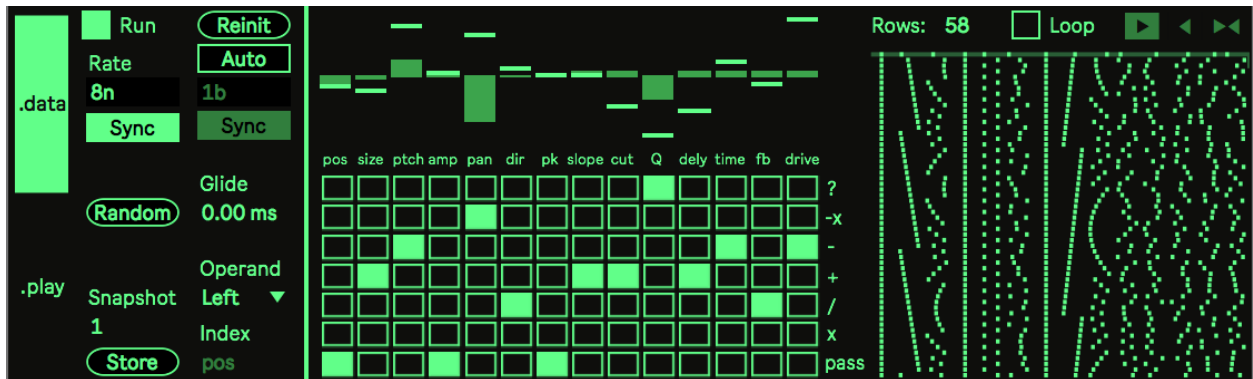


data.train User Manual



To install:

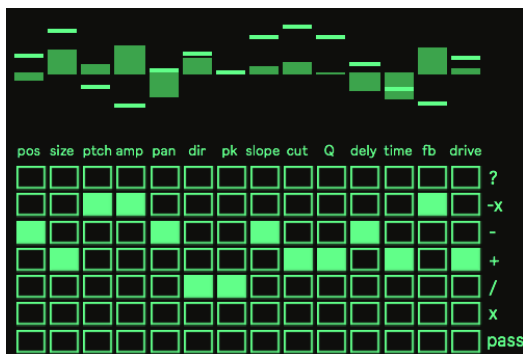
Unzip the folder and drop the folder called “data.train” in this exact location in order for presets to load correctly: *ableton/user library/presets/audio effects/max audio effect*

For best results or if you are having issues, make sure you are using the latest version of max/msp. You do not need to have a license if you are using Live suite. Download the newest version here: <https://cycling74.com/downloads> and once downloaded go to the ableton Preferences > Library and set the newly downloaded version of max to the one ableton should use. (Also you can try to see if it works fine with your bundled version first).

Synopsis

data.train is a max for live audio effect, but can also be used as an instrument. It is essentially a sampler mangler. You can either drop a sample file or record audio into the effect to mangle your sounds. How it “mangles” the sample playback/fx is through modulating the parameters via an ever changing array of values in the “data” section. Each frame, these values are changed via a selection of operations. Depending on the initial values and the selected operations, these values can produce various types of oscillations, from simple to complex. Because of the various synchronized timing options, this device is great for rhythmic effects, but also can be used for many other ones as well including glitchy, ambient, droney, etc.

THE “data” SECTION



The **multisliders** on top show the modulation values of each parameter in the device. The bar multislider set the original modulation values (from -1 to 1). These are also called the “seed” values. The brighter green horizontal bars layered over them display the current modulation values. Below each multislider is a label of the parameter that it modulates. “?” gives a random value each frame, “-x” negates the value, “-” is subtraction, “+” is addition, “/” is division, “x” is

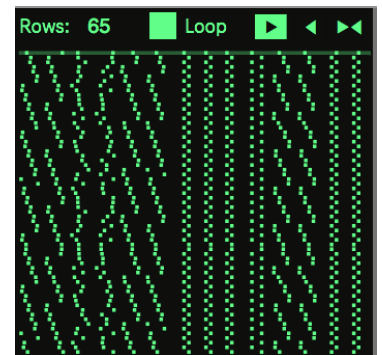
multiplication, and “pass” keeps the same value. The button **matrix** below the multislidiers set what operation the value undergoes each new frame. Each column of the matrix is associated with the labelled parameter, while each row relates to an operation which is listed to the right side of the matrix. For example, in the picture above, the loop “size” modulation undergoes an addition (+) operation every frame, which means its current value is added to another value to get the next modulation value. The value which it is added to is determined by the Operand parameter, which is described below.



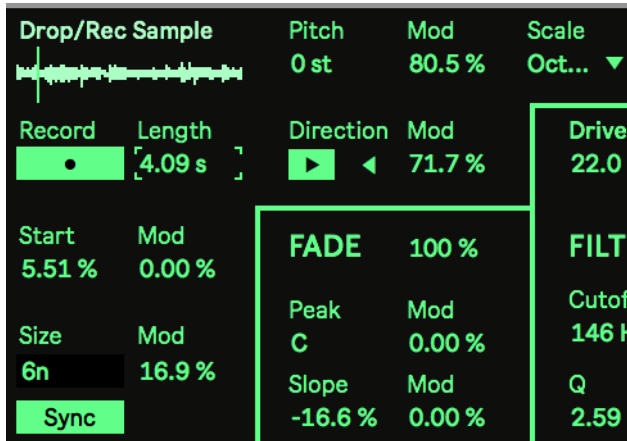
The leftmost side of this section controls the settings of the frame processing of the data. **Run** starts/stops the data processing. You can set the **Rate** in milliseconds or clock **synced** intervals. **Reinit** resets the modulation values to the original “seed” values. You can also **Auto** reinit the values either in milliseconds intervals or clock **synced** intervals. **Glide** allows for gliding between new modulation value changes for smooth transitions. You can also **Randomize** all of the parameters in the “data” section and store/recall **Snapshots** of all values in this section. The **Operand** parameter sets which value becomes the right hand operand in the value operation on each frame. “Left” sets the operand to the value left of the current value, “Right” sets the value to the value to the right of the current value, “Op” sets the right hand

value to the last value that went through the same operator, “Self” sets the right hand value to itself, “Index” sets the value to a specific value from the list set by the **Index** parameter below. So, following our example described in the paragraph above about the loop size modulation with the addition (+) operator... if Operand is set to left, then the loop size modulation value will be added with the start “pos” modulation value to get its new value. For Right, it would be added with the “pitch” modulation value, for Op it would be added with the “drive” modulation value, as that is the last value to also go through the addition (+) operator in the example picture. For Self it would be added to itself, and for Index it would again be added with the start “pos” modulation value as that is the selected Index parameter in the example picture in the above left.

In this section a history of frames of modulation values is visualized based on the **Rows** parameter. If **Loop** is off, this history does nothing functionally other than show the changes of the modulation values. If Loop is on, then these values are looped and repeated instead of new values being calculated. The arrows to the right of Loop set the **direction** of looping (forward, backward, foreback).

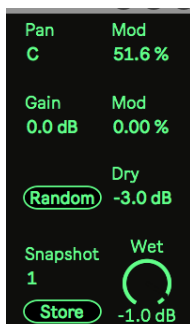
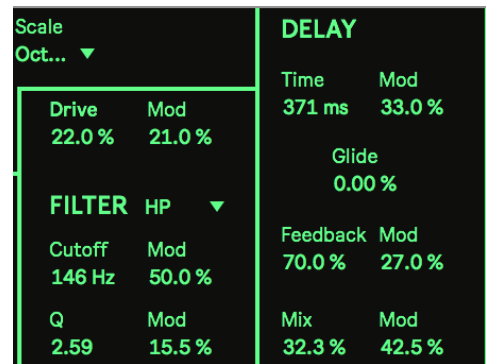


THE “play” SECTION



This section controls the sample playback and effects as well as the amount of modulation applied to each parameter. Every parameter labelled “Mod” is the amount of modulation that can be applied from the “data” section to the right of its associated parameter. In the top left is a display of the current sample. You may drop a sample file on that display to load it for playback. Alternatively, you may just **Record** audio for the inputs to the buffer and set the **Length** of the buffer as well. Below this you can set the **Start** position of the sample playback and the

Loop size either in milliseconds or clock **synced** intervals. Also in this section you can transpose the **pitch** (which also changes the speed) of playback, if modulation is applied to the pitch you have an option of applying quantized pitch intervals according to **Scales** (a scala file, .scl, can be dropped on this menu in order to load custom tunings and scales!). You can also change the **Direction** of playback. **FADE** controls the amount of window fading at the loop ends. **Peak** sets the center of that fade window. With fade at 100% and peak in the middle, the window is a triangle, with peak all the way to the left it is a ramp down, and peak all the way right is a ramp up (soft attack). **Slope** applies an exponential slope to make the window more “open” or “closed”. There are some effects that can be applied and modulated as well. An **Overdrive**, a basic biquad **Filter**, and a **Delay** line. The **Glide** parameter allows for gliding of the changes to the delay **time**. If at 0% glide, there are no pitch changes when changing the delay time which is obtained through rapidly crossfading multiple delay lines. With glide above 0%, there are gliding pitch changes in time changes.



Lastly are some basic mix options. **Panning** and **Gain** modulation can be applied here. There are separate volumes for the **Dry** signal (the signal coming into the device) and the **Wet** signal. You can also **Randomize** the parameters into his “play” section as well as store/recall **Snapshots** of them!

I hope you enjoy this device! Please email me if you have bugs or other issues:

dillonbastan@gmail.com

More: <http://dillonbastan.com>