

AGGREGATE MUSIC SYSTEM

OPERATION MANUAL

VERSION 1.0.0



AGGREGATE

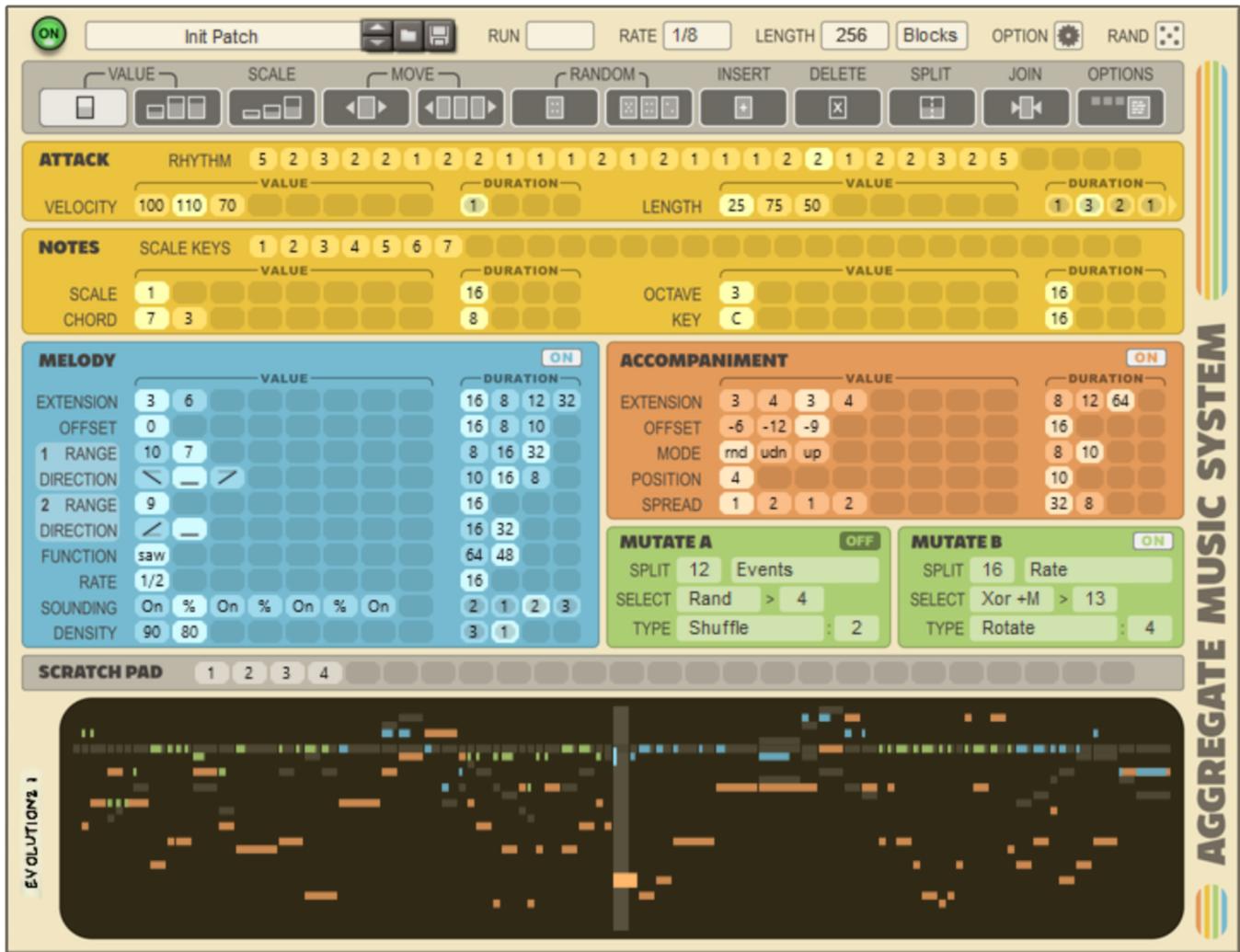
ag·gre·gate | \ `a-gri-, gāt \

aggregated; aggregating

transitive verb

1: to collect or gather into a mass or whole

2: to amount to (a whole sum or total) : TOTAL



INTRODUCTION

AGGREGATE is a generative music player for Reason that allows the user to control meta-parameters of music composition. Lists of values are iterated over with time durations then aggregated with other lists such that they change over time in complex ways. Users define a rhythm, scale, and chord progression which are then used as the basis for two separate generators, a melody generator and an accompaniment generator. The musical results can then be further mutated with two separate mutation engines.

Many aspects of this system are rooted in the works of Joseph Schillinger (1895–1943). His system of musical composition is based on iteration and permutation. The core of his methods have been expanded upon with this player.

AGGREGATE is the result of half a decade of coding and experimentation. The core generative engine was written and re-written multiple times using multiple programming languages and multiple platforms. Each new version improving and refining its methods. Fragments of the system were released in the smaller Rack Extension **EVOLUTION**.

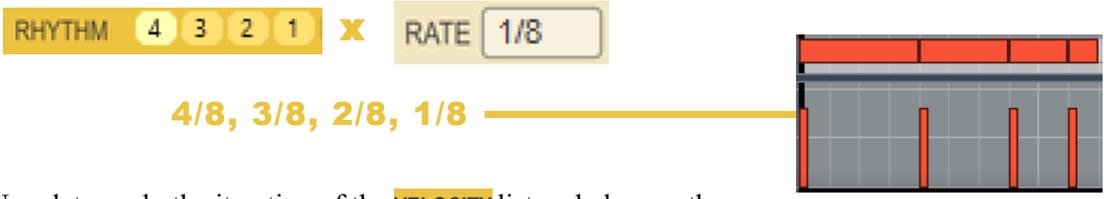
AGGREGATE is the newest form of this process.

EVENTS

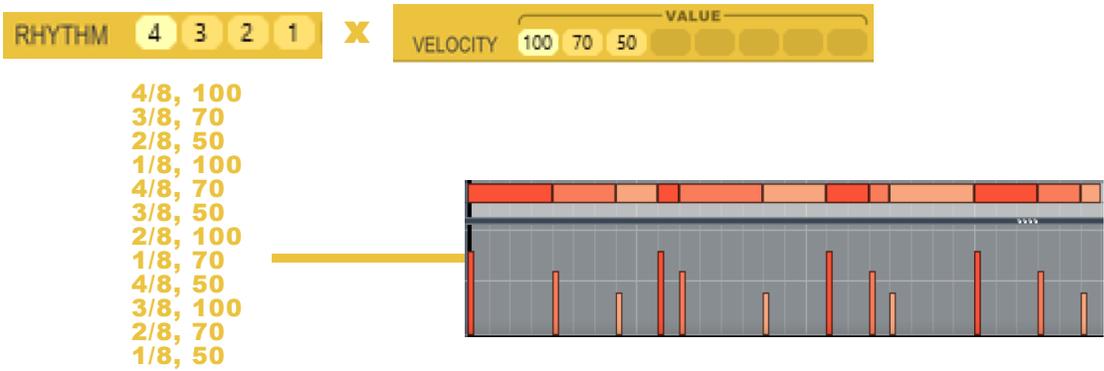
AGGREGATE uses the concept of events. An event is something that takes place for a duration of time. An event ends up as a single MIDI note, multiple MIDI notes, or a rest period. The origin of **all** events starts with the **RHYTHM** list in the **ATTACK** section.



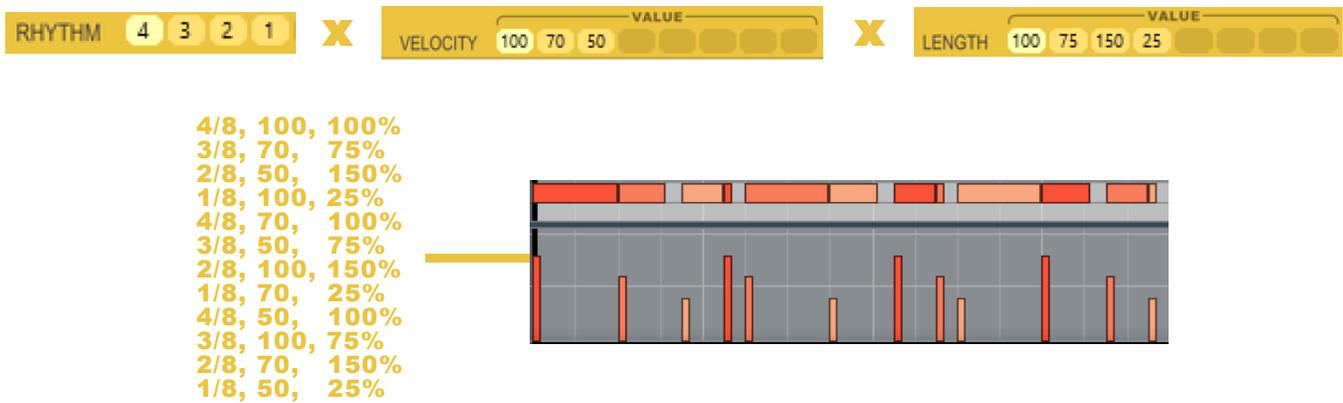
This generates a stream of events, each with the given rhythmic duration. This duration is a multiple of the global **RATE**. Applying a **RHYTHM** and a **RATE** we get:



Now lets apply the iteration of the **VELOCITY** list and observe the result.



Taking it further with **LENGTH**, which effects the note length as a percentage of rhythmic duration.



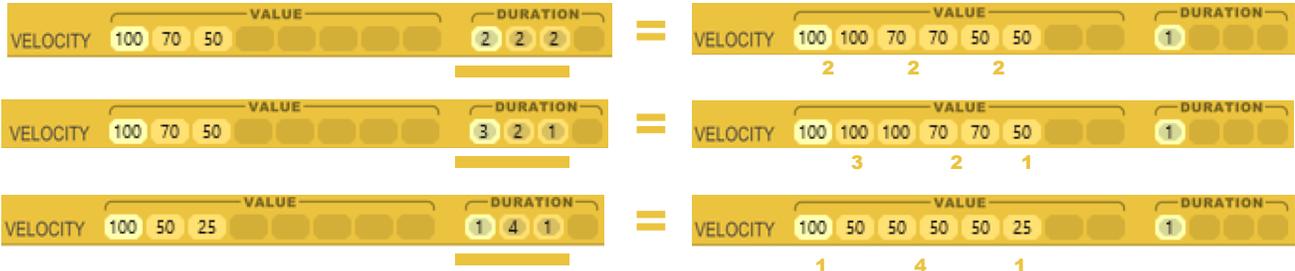
All the lists work in this manner to build up an event and eventually assign all MIDI note parameters.



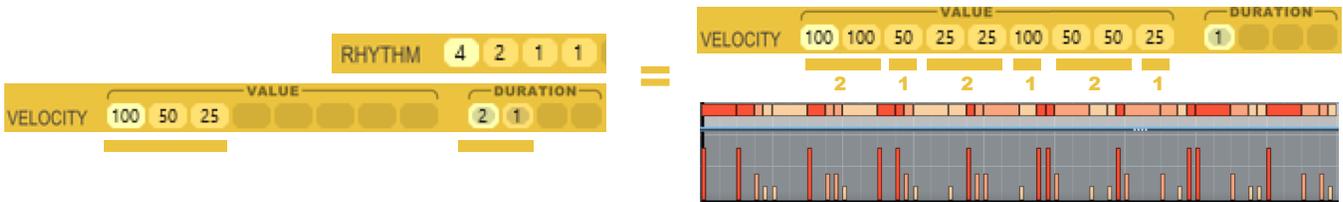
DURATION

You may have noticed the **DURATION** lists next to each value list. These sections give us a little more flexibility in how our values are iterated over and applied to the event. Instead of a strictly one-to-one pairing of list values, we can pair up our values multiple times before moving on to the next value. There are two types of durations, **Event** and **Rate**.

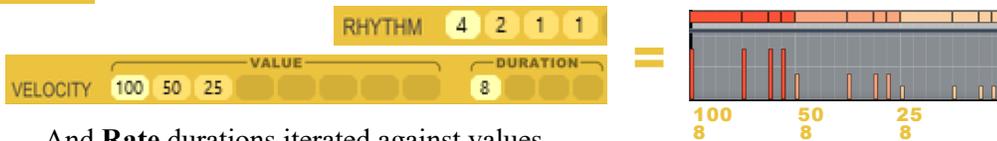
An **Event** type of duration is just a count of how many times a value is repeated before moving on to the next. The following **Event** type durations are *equivalent*.



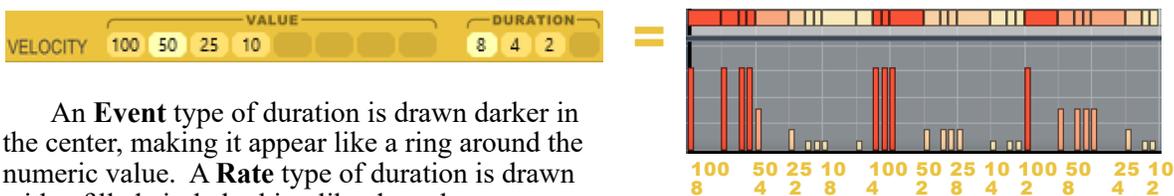
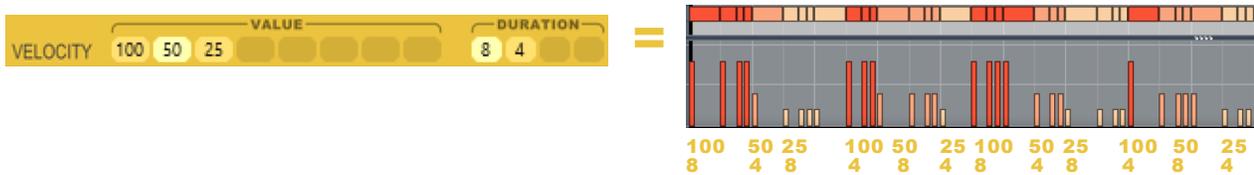
Durations are *also* iterated against the values, which makes for further interesting permutations.



A **Rate** type of duration specifies the amount of musical time a value will persist. For instance, if the global **RATE** is 1/8, the duration of 8 would mean 8/8 or a bars worth of that value. Below is 3 bars of **VELOCITY** iteration.



And **Rate** durations iterated against values.



An **Event** type of duration is drawn darker in the center, making it appear like a ring around the numeric value. A **Rate** type of duration is drawn with a filled circle looking like the values.

TOOLS



There is a palette of tools available to manipulate the lists. Select the tool by clicking on it. The **SCRATCH PAD** is a good place to practice using these tools.

VALUE

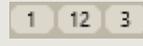
The **VALUE** tools let us change one or all items. Click dragging up or down on the list will increase or decrease the values.



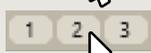
Single



Click Drag Up + 10 →



All

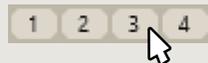


Click Drag Up + 10 →



SCALE

The **SCALE** tools let us adjust all list items proportionally. Click dragging up or down on the list will multiply or divide all the values.



Click Drag Up x 2 →



MOVE

The **MOVE** tools let us shift one or all items. Click dragging left or right on the list will move the position of the value. Items will wrap around the list.



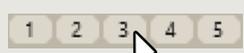
Single



Click Drag Right + 2 →



All



Click Drag Right + 2 →



TOOLS CONT.

RANDOM

The **RANDOM** tools let us randomize one or all items.



Single

1 2 3

Click →

1 39 3

Click →

1 -7 3



All

1 2 3

Click →

12 112 -67

Click →

21 12 59

INSERT

The **INSERT** tools let us add a new item to a list. The new item will have the same value as the clicked item.



1 2 3 4

Click →

1 2 3 3 4

DELETE

The **DELETE** tool will remove an item from the list.



1 2 3 4

Click →

1 2 4

SPLIT

The **SPLIT** tool will split a list item into two and divide up the value between them.



8 8 8 8

Click →

8 8 4 4 8

JOIN

The **JOIN** tool will merge two list items together and add their values.



8 8 8 8

Click →

8 16 8

OPTIONS MENU

OPTIONS

The **OPTIONS** tool will pop-up a context menu of choices to manipulate the list.



- Insert
- Delete
- Split
- Join
- Rand Single
- Rand All
- Set Value >

Perform the same actions on the list as the dedicated tools.

Shows a list of common values for the list type.

- Duration Type >

Some lists have additional options that effect how their values are interpreted.

Duration Type and *Chord Progression Type* are two examples.

- Expand & Permute >

Produces a list of different ways to manipulate the entire list. Permutations will slice the list into chunks.

- 36
- 30
- 24
- 18
- 12
- 6
- 0
- 6
- 12
- 18
- 24
- 30
- 36
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 12
- 14
- 15
- 16
- 18
- 20
- 22

- Axis 1
- Axis 2
- Average
- Min
- Max
- Interlace
- Cosine
- Sine
- Saw
- Square

- Reverse Abc > cbA
- Repeat Abc > AbcAbc
- Interlace Abc > AAbbcc
- Mirror Abc > AbccbA
- Permute 2 Abcd > Abcd cdAb
- Permute 3R Abc > Abc cAb bcA
- Permute 3L Abc > Abc bcA cAb
- Permute 4R Abcd > Abcd dAbc cdAb bcdA
- Permute 4L Abcd > Abcd bcdA cdAb dAbc
- Shuffle Abcd > cbAd
- Shuffle+ Abcd > bbcdAcdbcdAbcAAAdAbcd

OPTIONS MENU CONT.

Sequence Families >

Set the list to values that are a good source for rhythmic generation. Use on the **SCRATCH PAD** as parameters to other functions.

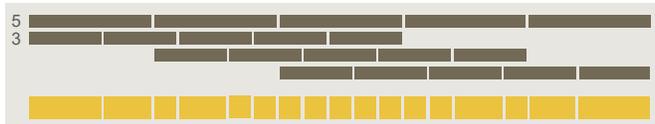
Interfere >

Overlay the durations of each base rhythm on top of each other to produce a new rhythm.



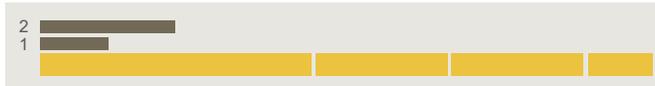
Fraction >

Overlay the durations of base rhythm **x** with multiple copies of **y**, each **y** copy resynced at the next **x**.



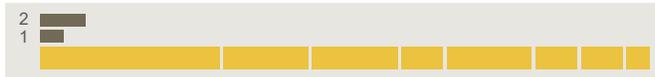
Square >

Polynomial expansion of $(XY)(XY)$ without simplification. (XX,XY,YX,YY) .



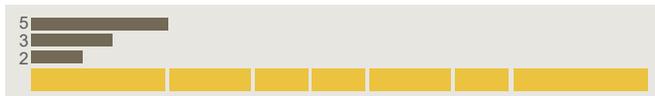
Cube >

Polynomial expansion of $(XY)(XY)(XY)$ without simplification. $(XXX,XXY,XYX,XYX,YXX,YXY,YYX,YYY)$.



Random >

x, **y** and **z** chosen randomly.



Copy
Paste

Copy, Paste this will copy and list to, and paste from the **SCRATCH PAD**.

12 3 5 8 13 21
13 4 7 11 18 29
14 5 9 14 23 37
15 6 11 17 28 45
16 7 13 20 33 53
17 8 15 23 38 61
25 7 12 19 31 50
27 9 16 25 41 66
35 8 13 21 34 55
37 10 17 27 44 71

32
43
52
53
54
65
72
73
74
75
76
83
85
87
92
94
95
97
98
85
87
92
94
95
97
98
94
95
97
98
12

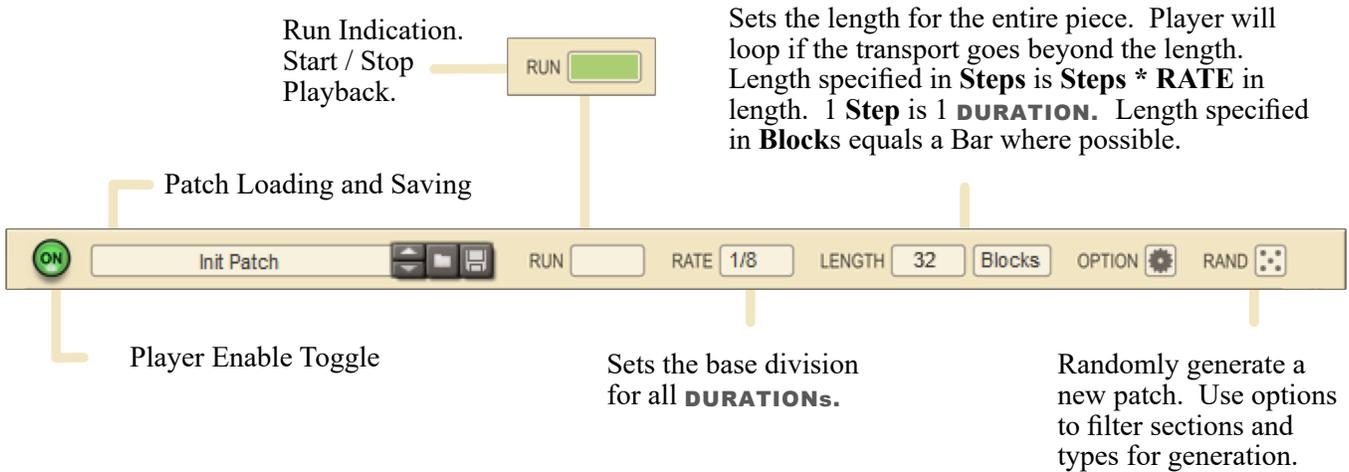
21
31
32
41
43
51
52
53
54
61
71
72
81
211
221
223
225
311
331
332
411
441
511
611
711
12
123

21
32
31
43
41
53
52
74
73
71
85
83
82
81
95
94
91
321
431
541
532
521
743
731
853
832
821
954
941
12
123

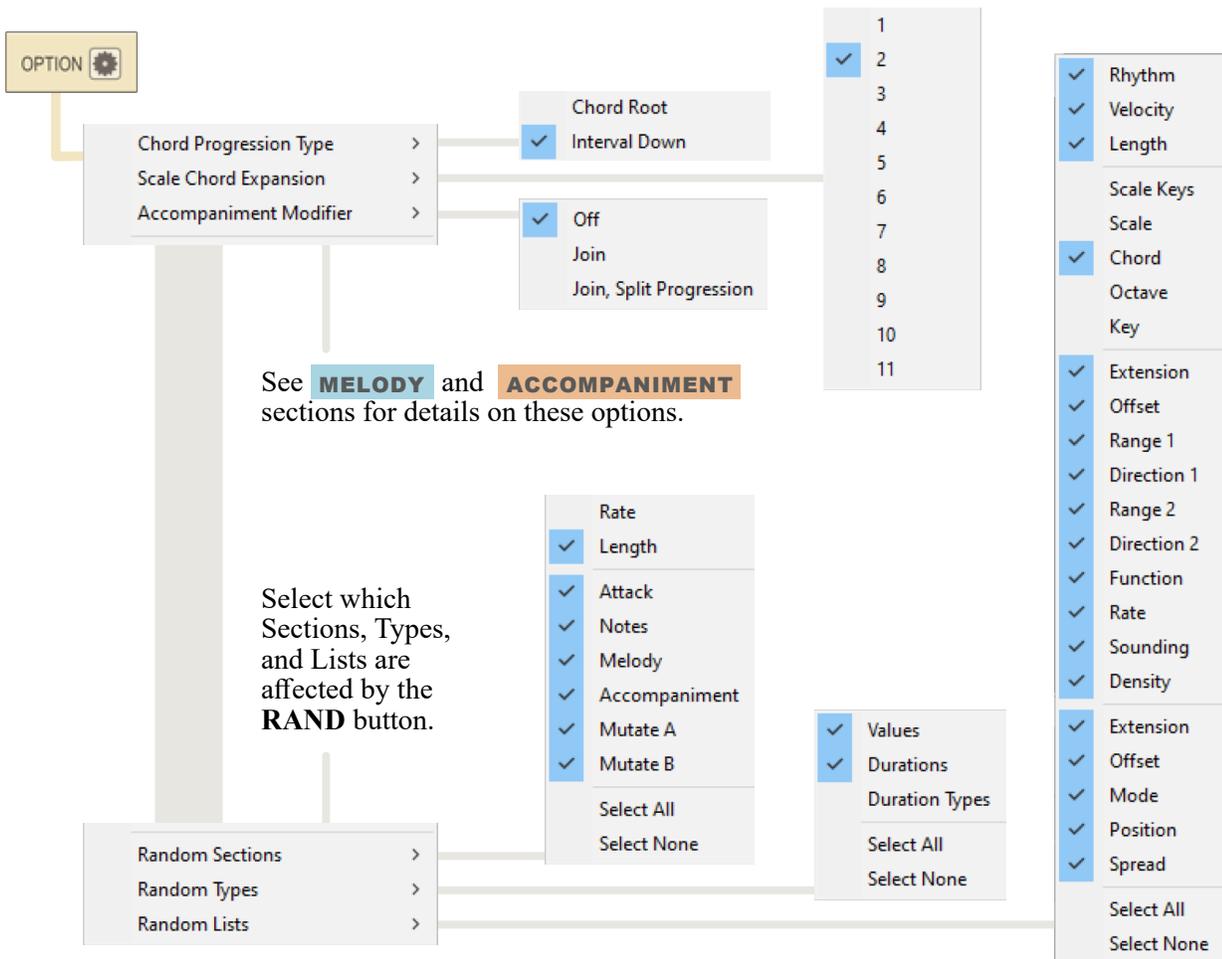
Last parameters on the menu are the first few values currently on the **SCRATCH PAD**.

1 2 3

TOP PANEL



MAIN OPTIONS



ATTACK

The screenshot shows the ATTACK control panel with the following settings:

- RHYTHM:** 3 2 1 2 1 1 1 1 1 1 1 1 2 1 2 3
- VELOCITY:** 100 110 120 50 90 60
- LENGTH:** 100 75 50 150 200 75

The **ATTACK** section is responsible for setting the foundation rhythm, pulse, and meter of the musical piece.

RHYTHM sets the duration for each individual event. The section will repeat until the total **LENGTH** has been reached.

The diagram shows the RHYTHM control set to 4 3 2 1, followed by a sequence of red bars representing the duration of each event.

VELOCITY sets the attack velocity.

The diagram shows the VELOCITY control set to 100 70 50, followed by a sequence of vertical bars of varying heights representing the attack velocity.

LENGTH sets the length of the final note as a function of the base **RHYTHM** duration.

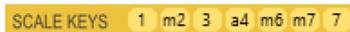
The diagram shows the LENGTH control set to 100 75 150 25, followed by a sequence of red bars of varying lengths representing the length of the final note.

Consult the **EVENTS** and **DURATION** section for more information on how these lists are combined together.

NOTES



The **NOTES** section is responsible for selecting our scales, keys, and chords. It is much more than just your basic note quantizer. These elements changing over time create the scaffolding of available notes for the **MELODY** and **ACCOMPANIMENT** generators.



SCALE KEYS defines scales in terms of note intervals. A Scale can be as little as one note, up to the chromatic scale with all 12 notes. There is a special menu when using the **OPTIONS** tool that sets a pre-defined scale.

Multiple scales are defined with the scale separator **:**. This will split them up for selection by the **SCALE** values.



SCALE selects the active **SCALE KEYS** scale.



OCTAVE and **KEY** control the musical center of the generators. The **MELODY** and **ACCOMPANIMENT** are anchored to this note. This would also define the root note of the 1 **CHORD**.

Set Scale

- Major
- Harmonic Major
- Minor
- Harmonic Minor
- Melodic Minor Ascending
- Melodic Minor Descending
- Ionian
- Dorian
- Phrygian
- Lydian
- Mixolydian
- Aeolian
- Locrian
- Acoustic
- Altered
- Enigmatic
- Flamenco
- Super Locrian
- Double Harmonic
- Gypsy
- Half Diminished
- Hungarian Minor
- Hungarian Major
- Lydian Augmented
- Major Locrian
- Neapolitan Major
- Neapolitan Minor
- Persian
- Phrygian Dominant
- Ukrainian Dorian
- Major Pentatonic
- Minor Pentatonic
- Hirajoshi
- In
- Insen
- Iwato
- Yo
- Augmented
- Blues
- Prometheus
- Harmonics
- Tritone
- Tritone Two-Semitone
- Whole Tone
- Octatonic Symmetric Whole
- Octatonic Symmetric Half
- Chromatic

NOTES CONT.

CHORD 5 3

CHORD has two modes of operation, **Chord Root** and **Interval Down**. These are accessible via the **OPTIONS** tool or the main **OPTIONS** menu. The **Scale chord expansion** is selectable as well.

Chord Root is your typical numbered chord progression. 1, 4, 5 would equate to chords I, IV, V.

Interval Down is a cyclic progression downward.

A cycle of 3rds would give 1,6,4,2,7,5,3,1. A cycle of 5ths would give 1,4,7,3,6,2,5,1. And a cycle of 7ths would give 1,2,3,4,5,6,7,1. *Many* popular chord progressions can be written as a combination of 3rd, 5th, and 7ths down. Using this method with values 3,5,7 produces many musically pleasing chord progressions.

Scale Chord Expansion is the means of building the chords from a scale. Typical chords are build with an expansion of 2. Meaning, we take every 2nd note from the chord root:

Scale 1 2 3 4 5 6 7 1 2 3 4 5 6 7
 Chord 1 : 1 3 5 7 2 4 6
 Chord 2 : 2 4 6 1 3 5 7

We are free to select other **Scale Chord Expansions** such as 1:

Scale 1 2 3 4 5 6 7
 Chord 1 : 1 2 3 4 5 6 7
 Chord 2 : 2 3 4 5 6 7 1

Or perhaps 3:

Scale 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7
 Chord 1 : 1 4 7 3 6 2 5
 Chord 2 : 2 5 1 4 7 3 6

CHORDS are used by the **MELODY** and **ACCOMPANIMENT** generators. Some or all of the chord notes are selected with the **EXTENSION** and **EXTENSION** list.

MELODY

MELODY		VALUE				DURATION		ON		
EXTENSION	3	4	6	4			16			
OFFSET	0						64			
1 RANGE	14	13					8	64		
DIRECTION	↙						8			
2 RANGE	14	6					64			
DIRECTION	↔	↗					32	16	8	16
FUNCTION	avg	a2	cos				64	16	48	
RATE	4	1/32	4				64			
SOUNDING	On	On	%	On	%	On	%	1		
DENSITY	90	60	90	40	90		1	2		

EXTENSION 3 4 6 4

EXTENSION selects the number of chord notes the melody is based on. A value of 1 will only use the root note of the chord when generating the melody. A value of 3 would use the basic chord triad. A value equal to the number of scale notes would allow the full scale to be used.

OFFSET 0

OFFSET is the number of semitones we are offset from the base **OCTAVE** and **KEY**. This is the position that axes 1 and 2 are anchored. This could also be thought of as axis 0.

1 RANGE 14 13 2 RANGE 14 6
DIRECTION ↙ DIRECTION ↔ ↗

There are two axes defined by **RANGE**, **DIRECTION**, and their **DURATIONS**. The axes define movement away from and toward the baseline **OFFSET**.

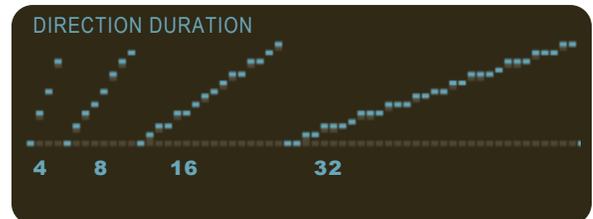
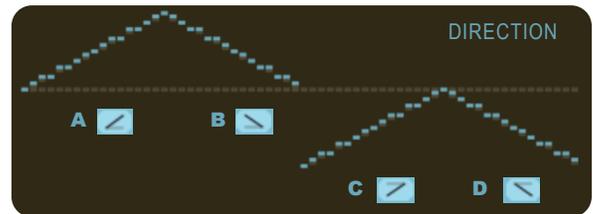
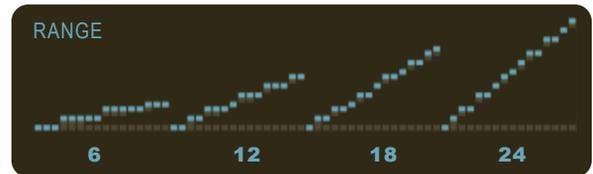
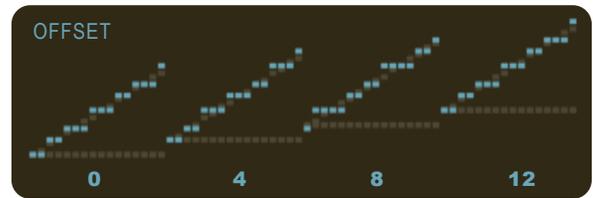
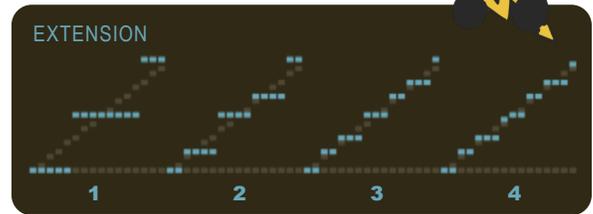
RANGE is the number of semitones away from the baseline, or the height of the shape.

DIRECTION is the trajectory in regards to the baseline.

- A moving up and away from the baseline.
- B moving down and toward the baseline.
- C moving up and toward the baseline.
- D moving down and away from the baseline.
- 0 on the baseline.
- 1 full range away from the baseline.

DIRECTION's **DURATION** defines the width of the shape.

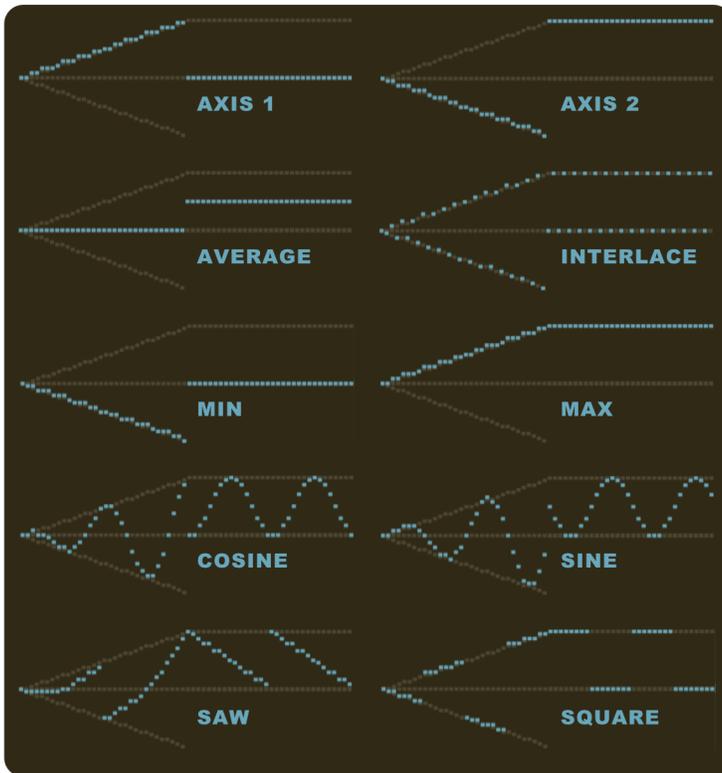
The **MELODY** section generates a monophonic melody using the rhythmic events from the **ATTACK** section and the scales and chord progressions defined in the **NOTES** section. A melodic path is defined by the offset, axis 1, and axis 2. We can further apply a mathematical function to those two axes.



MELODY CONT.

FUNCTION avg a2 cos

FUNCTION lets you select which axis to use, or defines a mathematical function bounded by both axes.



AXIS 1 only axis 1.

AXIS 2 only axis 2.

AVERAGE average of axes 1 and 2.

INTERLACE alternate axes 1 and 2 every event.

MIN minimum of axes 1 and 2.

MAX maximum of axes 1 and 2.

COSINE cosine function bounded by axes 1 and 2.

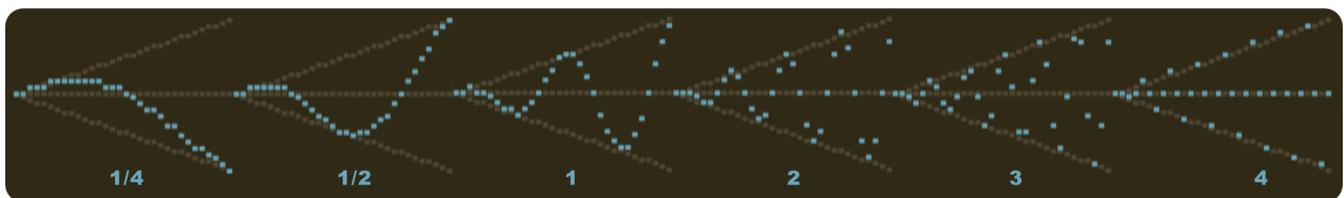
SINE sine function bounded by axes 1 and 2.

SAW saw function bounded by axes 1 and 2.

SQUARE square function bounded by axes 1 and 2.

RATE 4 1/32 4

RATE adjusts the function rate for cosine, sine, saw, and square functions.



MELODY CONT.

SOUNDING On On % On % On

SOUNDING sets the state of the note.

ON the note is on. (*if velocity from attack is not zero*)

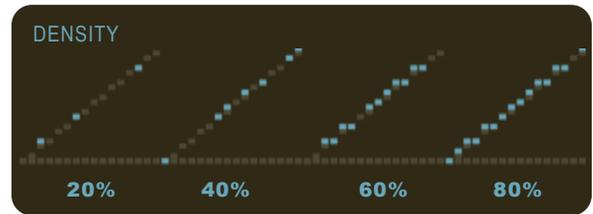
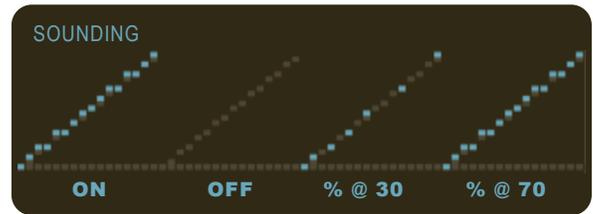
OFF the note is off.

% the note's on/off state depends on **DENSITY**.

One intended use of this parameter is to override **DENSITY**. It allows you to have low density melodies, but keep one section, or every Nth note ON to keep a rhythm.

DENSITY 90 60 90 40 90

DENSITY is the percentage of notes that are **ON**.



ACCOMPANIMENT

ACCOMPANIMENT		VALUE				DURATION		ON
EXTENSION	4	3			8	16		
OFFSET	-8	-11	-6			16		
MODE	dn	udn	rnd	dn		32	16	12
POSITION	fit	4				10		
SPREAD	2					64		

EXTENSION 4 3

EXTENSION selects the number of chord notes the final chord is based on. A value of 1 will only use the root note of the chord when generating the melody. A value of 3 would use the basic chord triad. A value equal to the number of scale notes would allow the full scale to be used.

OFFSET -8 -11 -6

OFFSET is the number of semitones we are offset from the base **OCTAVE** and **KEY**. This is the position that the chord center will gravitate towards by shifting octaves.

MODE dn udn rnd dn

MODE determines what kind of accompaniment is produced. Notes are limited to those in the **EXTENSION**.

FULL CHORD All the notes in the **EXTENSION**

- | | |
|-----------------------|---------------------|
| 1 Root Note | 24 2 and 4 |
| 2 2 Note of the Chord | 34 3 and 4 |
| 3 3 Note of the Chord | 123 1, 2, and 3 |
| 4 4 Note of the Chord | 134 1, 3, and 4 |
| 12 1 and 2 | 124 1, 2, and 4 |
| 13 1 and 3 | 234 2, 3, and 4 |
| 14 1 and 4 | 1234 1, 2, 3, and 4 |
| 23 2 and 3 | 2+ 2 and up |

ARP UP Arpeggiate Up

ARP DOWN Arpeggiate Down

ARP UP+DOWN Arpeggiate Up and Down

ARP RAND Arpeggiate Random

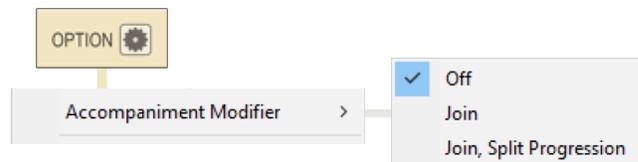
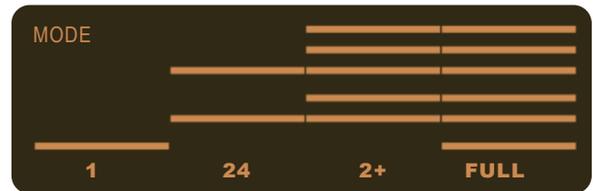
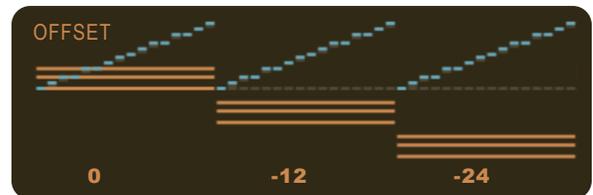
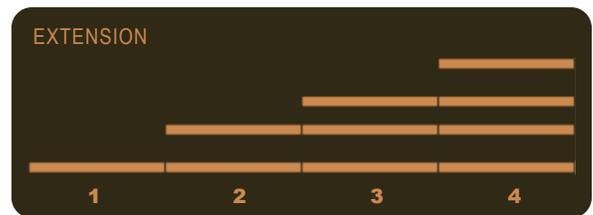
ACCOMPANIMENT MODIFIER can be used to merge events when the chords match.

OFF Default, no merge.

JOIN Join any events that have the same notes.

JOIN, SPLIT PROGRESSION Join any events that have the same notes, but still split at **CHORD** duration.

The **ACCOMPANIMENT** section generates a polyphonic chord progression or monophonic arpeggiation using the rhythmic events from the **ATTACK** section and the scales and chord progressions defined in the **NOTES** section.



ACCOMPANIMENT CONT.

POSITION fit 4

POSITION determines the chord voicing position

1 Chord in root position.

2 - 12 Note in the lowest position.

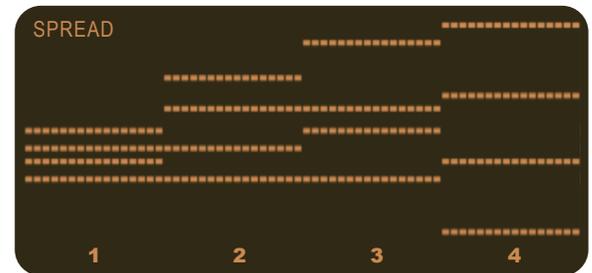
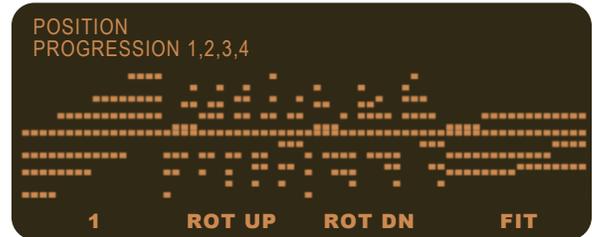
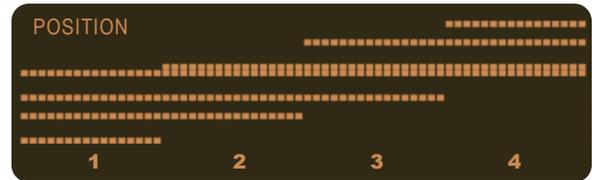
ROTATE UP Each event rotates the position up one.

ROTATE DOWN Each event rotates the position down one.

FIT CENTER Choose whatever position best fits the chord's gravity to the **OFFSET**

SPREAD 2

SPREAD controls the number of jumps between chord notes. This will cause the chords to widen out.



MUTATE



MUTATE A <input type="checkbox"/> ON	MUTATE B <input type="checkbox"/> ON
SPLIT 6 Events	SPLIT 8 Rate
SELECT Rand > 2	SELECT M:N > 2 : 7
TYPE Warp : 1	TYPE Warp : 1

The **MUTATE** sections chunk up the **MELODY** and **ACCOMPANIMENT** events and modify them in different ways. The **A** and **B** mutations are independent and run one after the other.

SPLIT determines how the events are split up into chunks. The types of split are the same as duration types, a number of **EVENTS** or a duration at **RATE**. See the **DURATION** section for more.

SELECT determines which chunks will have the mutation applied.

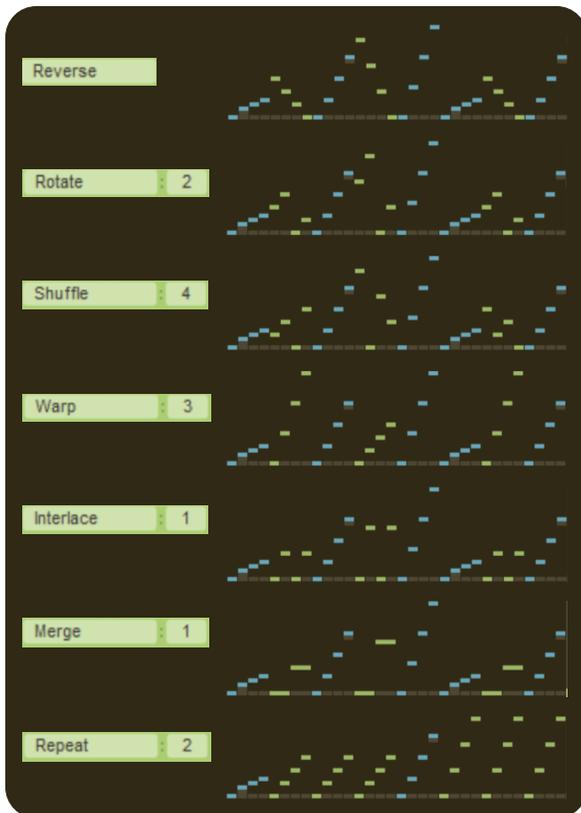
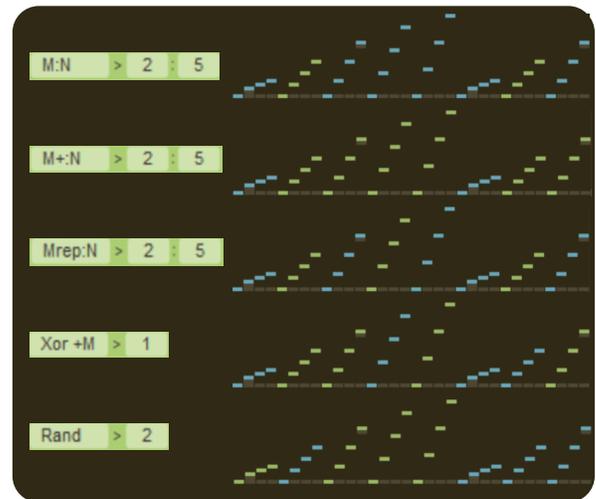
M:N Select the Mth chunk every N chunks.

M+:N Select the Mth to the Nth chunk every N chunks.

MREP:N Select every Mth chunk, reset every N chunks.

XOR+M Select every chunk N, where all the bits of the number N+M XOR'd together are 1.

RAND Select random chunks.



TYPE is the mutation to be applied to the selected chunks.

REVERSE Reverses all events.

ROTATE Shifts the events by X.

SHUFFLE Random shuffle of events, seeded with X.

WARP Replace events with events from chunk in position +X.

INTERLACE Iterate over chunked events skipping every X.

MERGE Merge every X events duration.

REPEAT Repeat the whole chunk X times.

THANKS

Special thanks to all the reasontalk.com beta testers and patch creators.

Bes
Catblack
EpiGenetik
Loque
MIDISwede
MarkTarlton
Mataya
MrFigg
Noise
Ottostrom
Timmy Crowne
WaxTrax
Zac
artotaku
cgijoe
challism
deeplink
doze
freshkidblaze
joeyluck
kbard
loopeydoug
martynx
mjxl
olive6741
outdatedstandard
phobic
rbooth
retreed
riemac
strangers
tiker01

