

# The Lazy Producer - Starter ideas for generative music

Mike K Smith



Cover image by Danielle Navarro (<https://djnavarro.net>)

Image used by agreement with the artist.

MikeKSmith's music can be found on Bandcamp at  
<https://mikeksmith.bandcamp.com>

# Chapters

	1
<b>Preface</b>	<b>2</b>
You shouldn't read this book . . . . .	2
But if you DO read this book, what should you expect? . . . . .	2
What do you mean, "Lazy Producer"? . . . . .	3
Lorem Ipsum . . . . .	3
Reduce, reuse, recycle . . . . .	4
What will I need to be able to use the ideas in this book? . . . . .	4
Why Ableton? . . . . .	4
Structure of this book . . . . .	5
Who am I? . . . . .	5
<b>1 What do we mean "Generative" music?</b>	<b>7</b>
Generative: . . . . .	7
<b>2 Recipe - Changing chords</b>	<b>9</b>
<b>3 Process - Note-wise probability</b>	<b>14</b>
<b>4 Process - Follow Actions</b>	<b>21</b>
4.1 Follow ACTIONS . . . . .	21
4.2 FOLLOW Actions . . . . .	23
4.3 Follow Action probabilities . . . . .	23
4.4 A Random Walk . . . . .	24
<b>5 Tools - MIDI tools: chords and added notes</b>	<b>25</b>
5.1 Ableton's Note Echo plugin . . . . .	25
5.2 Ableton's Random plugin . . . . .	26
5.3 Ableton's Note Length plugin . . . . .	27
5.4 Ableton's Scale plugin . . . . .	27
5.5 Ableton's Chord plugin . . . . .	28
5.6 Probability Gate . . . . .	29
5.7 Deviate . . . . .	30
5.8 Max for Live Tintinnabulator plugin . . . . .	31
5.9 Max for Live Schwartzonator plugin . . . . .	32
5.10 Max for Live Chord Generator plugin . . . . .	32

## *Chapters*

<b>6 Process - Routing MIDI and Audio</b>	<b>34</b>
6.1 Splitting MIDI for a single instrument across tracks . . . . .	36
<b>7 Recipe - Doing more with less using MIDI tools and plugins</b>	<b>37</b>
<b>8 Tools - Modulators</b>	<b>44</b>
8.1 Modulators in Ableton Live (general rule of thumb) . . . . .	44
8.2 LFO . . . . .	47
8.3 LFO Trio - a chaotic LFO . . . . .	48
8.4 Envelope MIDI . . . . .	48
8.5 Envelope Follower . . . . .	49
8.6 Shaper . . . . .	50
8.7 MIDI Expression Control . . . . .	51
8.8 Strange Mod . . . . .	52
8.9 Dispatch . . . . .	53
8.10 Deviate . . . . .	54
<b>9 Process - Tuning Parameters</b>	<b>56</b>
9.1 The Problem . . . . .	56
9.2 The Taste Test . . . . .	56
9.3 Waving, not droning . . . . .	57
9.4 Be random, but not <i>too</i> random . . . . .	57
9.5 Tune the parameters . . . . .	57
<b>10 Process - Balancing the unexpected and the predictable</b>	<b>58</b>
10.1 Evolving changes . . . . .	58
10.2 Taming randomness - gardening the random processes . . . . .	59
10.3 Deliberate use of repeating patterns - ostinato and predictable rhythmic parts	59
10.4 Repeat yourself . . . . .	60
10.5 <i>YOU</i> define the parts, the computer picks the order . . . . .	60
10.6 Contextualising randomness using humans . . . . .	61
10.7 Play with the “1” - mess with the listener’s perception . . . . .	61
<b>11 Recipe - Balancing the unexpected and the predictable</b>	<b>62</b>
<b>12 Process - Workflow: capturing generative ideas</b>	<b>66</b>
12.1 In the beginning . . . . .	66
12.2 Random processes are random . . . . .	67
12.3 How to set up Live sets - my experience . . . . .	67
<b>13 Tools - MIDI Generators</b>	<b>72</b>
13.1 Melodic Steps . . . . .	72
13.2 Bouncy Notes . . . . .	73
13.3 Dr Chaos . . . . .	74
13.4 Euclidean Sequencer Pro . . . . .	75
13.5 Less Concepts . . . . .	75

## *Chapters*

13.6 Turing machines . . . . .	76
13.7 MIDI Waves . . . . .	77
13.8 Midivolve . . . . .	78
13.9 SEEDS - Polymath and friends . . . . .	79
13.10 Others I haven't mentioned . . . . .	80
<b>14 Recipe - Using MIDI generators</b>	<b>82</b>
<b>15 Tools - Audio Generators</b>	<b>86</b>
15.1 Pinging resonators / resonant filters . . . . .	86
15.2 Tree Tone . . . . .	87
15.3 Harmonic Drone Generator . . . . .	88
15.4 Cellular Degradation . . . . .	89
<b>16 Tools - Creative use of audio effects</b>	<b>90</b>
16.1 The tools of the trade . . . . .	90
16.2 Hybrid Reverb . . . . .	90
16.3 Valhalla Supermassive . . . . .	91
16.4 Echo . . . . .	92
16.5 Saturation and Distortion . . . . .	93
16.6 Resonators . . . . .	94
16.7 Spectral processing . . . . .	95
16.8 Other notable mentions . . . . .	96
16.9 FX in series vs Effects Rack vs Return channels vs Separate tracks . . . . .	98
16.10 Use an Envelope with your effect . . . . .	98
<b>17 Tools - Extreme audio stretching with PaulStretch</b>	<b>100</b>
17.1 Why would you want to stretch audio? . . . . .	100
17.2 What does it do? . . . . .	101
17.3 How do I use it? . . . . .	101
17.4 What should I use it for? . . . . .	102
<b>18 Recipe - Making a drone using some audio generators and effects</b>	<b>104</b>
<b>19 Process - Using Scales</b>	<b>107</b>
19.1 Pentatonics - Minor and Major . . . . .	107
19.2 Hirajoshi . . . . .	108
19.3 Modes . . . . .	108
19.4 There are no “wrong” notes, only context . . . . .	109
19.5 Selections of notes - chords . . . . .	109
19.6 Other scales & traditions . . . . .	110
<b>20 Process - Using Field Recordings</b>	<b>111</b>
20.1 Balancing the natural and the (unnatural) randomness . . . . .	111
20.2 Natural and urban sounds . . . . .	111
20.3 Randomness does occur in “natural” sounds . . . . .	112

## *Chapters*

20.4 Recording noises . . . . .	112
20.5 Online sources for field recordings . . . . .	112
20.6 Using the samples . . . . .	112
<b>21 Recipe - Audio clip or Sampler into effects</b>	<b>113</b>
<b>22 Process - Unsynced loops</b>	<b>116</b>
22.1 Set phases to stunning . . . . .	116
22.2 Easy implementation in the DAW . . . . .	116
22.3 I'm in my prime . . . . .	117
22.4 Fibonacci and the Golden Ratio . . . . .	117
22.5 Let's ROTATE . . . . .	118
<b>23 Recipe - Using the Fibonacci sequence to create loops (after DivKid)</b>	<b>119</b>
<b>24 Tools - Live 12 Overview: MIDI and compositional tools</b>	<b>121</b>
24.1 Scale awareness . . . . .	121
24.2 Compositional tools . . . . .	123
24.3 Note probability . . . . .	125
<b>25 Tools - Live 12 Sequencers</b>	<b>127</b>
25.1 SQ . . . . .	127
25.2 Rhythmic Steps . . . . .	128
25.3 Step Arp . . . . .	129
<b>26 Tools - Live 12 Modulation</b>	<b>130</b>
26.1 Remote control or modulate? . . . . .	130
26.2 Live 12 - LFO shapes and modulation . . . . .	131
26.3 Live 12 - MIDI Expression Control . . . . .	133
26.4 Live 12 - Shaper . . . . .	133
<b>27 Tools - Live 12 MIDI Generation and Transformation</b>	<b>134</b>
27.1 MIDI Generation . . . . .	134
27.2 MIDI Transformation . . . . .	136
27.3 Limitations . . . . .	137
27.4 Extensions . . . . .	138
27.5 Philip Meyer MIDI Tools . . . . .	139



# Preface

## You shouldn't read this book

If you're looking for a guide to creating tracks from start to finish in a DAW then this is not that book.

If you're looking for a description of recording a band or producing genres outside of electronic or ambient music, this is not that book.

If you're looking for an in-depth guide to Ableton Live Suite, [then you should read the manual](#). I'm assuming you know your way around Ableton Live. Actually it's worth reading the manual anyway, as it's really well written and has a lot of useful information that is often overlooked.

If you are looking for generic tips on creating music I wholeheartedly recommend reading “Making Music: 74 Creative Strategies for Electronic Music Producers” by Dennis DeSantis<sup>1</sup>. This is a great book and will help you get through difficulties starting, continuing and finishing your compositions.

I also recommend Loopop’s “In-complete book of electronic music tips and tricks” available when you sign up as a Patron (<https://www.patreon.com/loopop>). His book contains an assortment of tips and tricks for making music including ideas for generative music.

## But if you DO read this book, what should you expect?

But if you're looking for some ways to generate fresh ideas to get you started and make some nice sounds that you could later build up into finished tracks, then this **IS** that book. What I'm going to try to present here is a collection of “starter ideas” or recipes for you to follow. I'll also talk a bit about some specific tools and plugins you might use, and touch on some general processes that I use in turning the techniques here into ambient, textural and generative music.

---

<sup>1</sup>DeSantis, D. *Making Music: 74 Creative Strategies for Electronic Music Producers*. Ableton AG, 2015. [https://cdn-resources.ableton.com/resources/uploads/makingmusic/MakingMusic\\_DennisDeSantis.pdf](https://cdn-resources.ableton.com/resources/uploads/makingmusic/MakingMusic_DennisDeSantis.pdf).

What do you mean, “Lazy Producer”?

## What do you mean, “Lazy Producer”?

Most professional musicians, music creators and producers I know are very hard working. They spend a LOT of time crafting a tune, song or mix with care and attention to making the final track as good as it can be. These individuals are most definitely, categorically NOT lazy.

The typical Ableton Live set I see from many electronic music producers has a huge number of tracks, precision edits, detailed automation, fills, transitions, risers, “ear-candy”. It’s a technicolor riot of clips. In stark contrast, my typical Live set for a track may have around 12 tracks. I use randomness, probability and automation to make things change from one bar or section to another. I start tracks largely by letting the computer make choices then tidy this, finding the “best” settings in the randomness and probability parameters to “garden” the generated random MIDI and audio to create a little more order. I am a Lazy Producer.

I will use the shorthand “Lazy Producer” in this book meaning that the techniques described will get you started on tracks, but are probably not finished tracks in themselves. However, if you’re a REALLY Lazy Producer, they might be. I have been known to do this.

## **Lorem Ipsum**

There’s a technique used in developing web pages and other documents where you can create placeholder text and images to fill space while you concentrate on layout, functionality of the page, the user experience and so on. This placeholder text is nonsense and is not there to be read by anyone, but it needs to take up the right amount of space and have the appearance of genuine content. It’s not enough to mash on the keyboard and write “reotiuyerwlgkjdfi ughweriugt hoierug oeriubgioeruhocr evhgieur hoieruhgoiuewrhggiuerhg coireuh voerwiugh oewrihg oiwrug gh” because that doesn’t have the right breakdown of syllables, spaces, sentences. Also, as the developer of the page you don’t want to have to take time to write anything to fill the space. As a consequence, developers rely on Lorem Ipsum text generators like the one at <https://www.lipsum.com>, to generate paragraphs of text. Here’s an example of a paragraph generated from the site mentioned.

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed varius gravida tortor nec dapibus. Donec at viverra dui, sed imperdiet ante. Integer ut dapibus leo. Quisque elementum tristique malesuada. Praesent venenatis elit eget mauris porta, in facilisis arcu aliquam. Phasellus ac leo tempus, pretium felis sit amet, sagittis nibh. Curabitur fermentum et dui in porttitor. Nulla et porta enim. Curabitur bibendum odio nisl, quis placerat lacus sodales quis. Vestibulum ut mollis urna. Mauris nec quam sit amet justo pretium aliquet.*

Many of the techniques I’m describing in this book are like Lorem Ipsum but for music. You can use them to generate content that is a placeholder in a production while you work on sound design, or to create musical or rhythmic ideas that you can bounce off in the creation of new parts. Then when the new idea is created it will be safe to delete the placeholder content and retain the good bits.

As a Lazy Producer, I often wind up keeping the Lorem Ipsum music if it sounds good.

## **Reduce, reuse, recycle**

It's natural as producers to keep an eye on developers producing sample packs, plugins and hardware and wondering if **THIS** product is going to make a difference to my next track. What I'd like to propose is that YOU can produce sample packs that work for YOU and the style of music you make much better than a third party developer. So while you work through the techniques in this book, [save the results and save the clips \(Ableton's Live Clips\) to your User Library](#), organising by type or feature, so that when you next need a new part or track starter, you can dip into this library and recycle what you have already prepared.

As a Lazy Producer I reuse MIDI parts that sound good across MANY tracks. This doesn't necessarily mean that many parts play EXACTLY the same MIDI part, but there are things that can be done with the same MIDI part that recycle it - making changes, but keeping parts of the original. I will demonstrate techniques for doing that in subsequent chapters.

## **What will I need to be able to use the ideas in this book?**

You will get most out of this book if you have Ableton Live Suite (from v11 onwards). Some of the ideas could be translated to other DAWs, but there are some features like note-wise probability which I'm not sure how to replicate in other DAWs.

Many of the tools and plugins discussed use Max For Live, which comes bundled with Ableton Suite. If you **don't** have Ableton Suite, then you can purchase Max For Live separately from [ableton.com](http://ableton.com). You **don't** have to purchase the full Max license from Cycling 74 to use Max For Live.

I'm not assuming that you know how to program with Max For Live - you only need to be aware that plugins we're talking about are programmed with that platform, so it needs to be available on your computer.

## **Why Ableton?**

I am a long time Ableton user (since Live 4). As I write this guide, I am using Live 12.2. I have a strong belief that Ableton Suite can be used much like a modular synthesiser - with the combination of audio and MIDI routing within Live, coupled with Max For Live devices, this amazing DAW can do so much. In fact, as you read through the chapters of the book you'll see how many of the tools, plugins and methods we'll talk about mimic what you would see and how you might use them for generative music in a modular synthesiser rack.

## **Structure of this book**

If you flick through the book you'll see different types of chapter.

### **Recipes**

Recipes chapters introduce a set of instructions that you can follow to create your own music using randomness and probability i.e. generative music. If you're a Lazy Producer like me then you can use these recipes to generate ideas and sounds that you can then use to start your own process of creating a track. If you're a lazy reader too, then you can skip straight to these chapters and follow the recipe. If you're a creatively Lazy Producer then you can tweak the recipes or combine them with other recipes to make your own creations. That's OK. Much like cooking recipes, you can adapt them to taste or preference, to whatever is in your cupboards (sound design wise) and depending on how much time you have to prepare and execute.

### **Process**

Process chapters talk about general ideas that can be applied within Ableton Live or potentially to other DAWs or music making processes. I'll try to steer clear of specific discussion of tools or plugins in these chapters.

### **Tools and plugins**

The chapters for geeks and musicians with Gear Acquisition Syndrome (GAS). If you want to get really nerdy or think about acquiring new plugins then these are the chapters for you. If you have already purchased Ableton Live then you have already invested a large amount on music making equipment. My goal here is **not** to persuade you to part with significant additional amounts of cash, but some of the plugins have an associated, although modest, cost.

### **Who am I?**

My name is Mike K Smith. I trained as a statistician and work as a data scientist in the pharmaceutical industry. I identify as a professional geek. Since making ambient music is a passion for me and doesn't have to pay the bills, I have the luxury of being a "Lazy Producer".

I make ambient, textural and sometimes generative music mostly using Ableton Live. I have released 3 albums collaborating with other musicians - where they prepared starter audio

*Who am I?*

“stems” of tracks which I then augmented and created finished tracks. I have also produced an album of drone and textural music under my own name.

You can find my music here:

<https://mikeksmith.bandcamp.com>

<https://soundcloud.com/mikeksmith>

# 1 What do we mean “Generative” music?

## Generative:

“Generative” music is where, following some trigger event such as pressing “play” or playing a note, or initiating some voltage (in modular synthesis rack), the music follows some rules, algorithms or uses probability to generate or evolve the musical ideas (notes, rhythms, timbres) for an arbitrary amount of time. In some cases the generative algorithms work *with* the performer, reacting to their input. In other cases the performer may intervene with the algorithms to tweak, adjust or change direction.

### 💡 Key idea

Generative music means (to me) that the music should be able to go on indefinitely, evolving over time rather than being simply repetitive.

In the context of generative *ambient* music, it should have enough going on to make it interesting, but without anything particularly “sticking out” to catch the listener’s ear.

Some generative algorithms take parts that the performer plays and augments this with additional harmony, chords or (counter) melodies. An example of this is [Olafur Arnalds' and Halldór Eldjárn's Stratus algorithm for performer pianos](#) where MIDI triggers generate sequences of associated notes, chords and “ripples” in a separate instrument or player piano.

Other generative music involves complex algorithms, probability triggers and modulations to create ever-changing sounds, sequences, rhythms. [Often these are programmed via modular synth rigs](#) using LFOs, sequencers, quantizers, harmonic generators, bernoulli (probability) gates, Turing machines (random note and sequence generators).

It’s your choice what you do with these generative parts - either shorter, reactive sequences or long infinitely varying sequences. Through modulation and automation it’s possible to create music where the sounds evolve, appear, disappear, blend or create dissonance. In this book we’ll try to present some ideas that will help get you started on this journey. I recommend that you try out the ideas in practice. Let the sequences play. Sit with them a while and let what you hear guide what to try next...If you like what you hear, save it. If you like parts of what you hear, save those as little self-contained ideas for use in other tracks. If you don’t like what you hear, then try replacing one or more parts and repeat

*Generative:*

the process of listening and refining. You can take the generative parts together to form a completely generative piece of music, or you can take only one element and build a track around that. Or you can take a snippet of one idea and use that as a kernel for new sound design, resampling it or reusing it in a completely structured and arranged piece of music.

This is a playground of choices - some ***you*** can make, and then some that are made within the generative processes.

## 2 Recipe - Changing chords

### Key idea

What we're trying to achieve here is to have clips launching out of phase with each other i.e. they don't all change at the same time. The clips also use probability on the chord notes, so that each time the clip plays we hear a slightly different selection of notes. Across the clips we're going to try to imply slightly different chords or tonal centres by changing the "root" of the chord in each clip. The combination of these elements should make a shifting pattern of chords and textures.

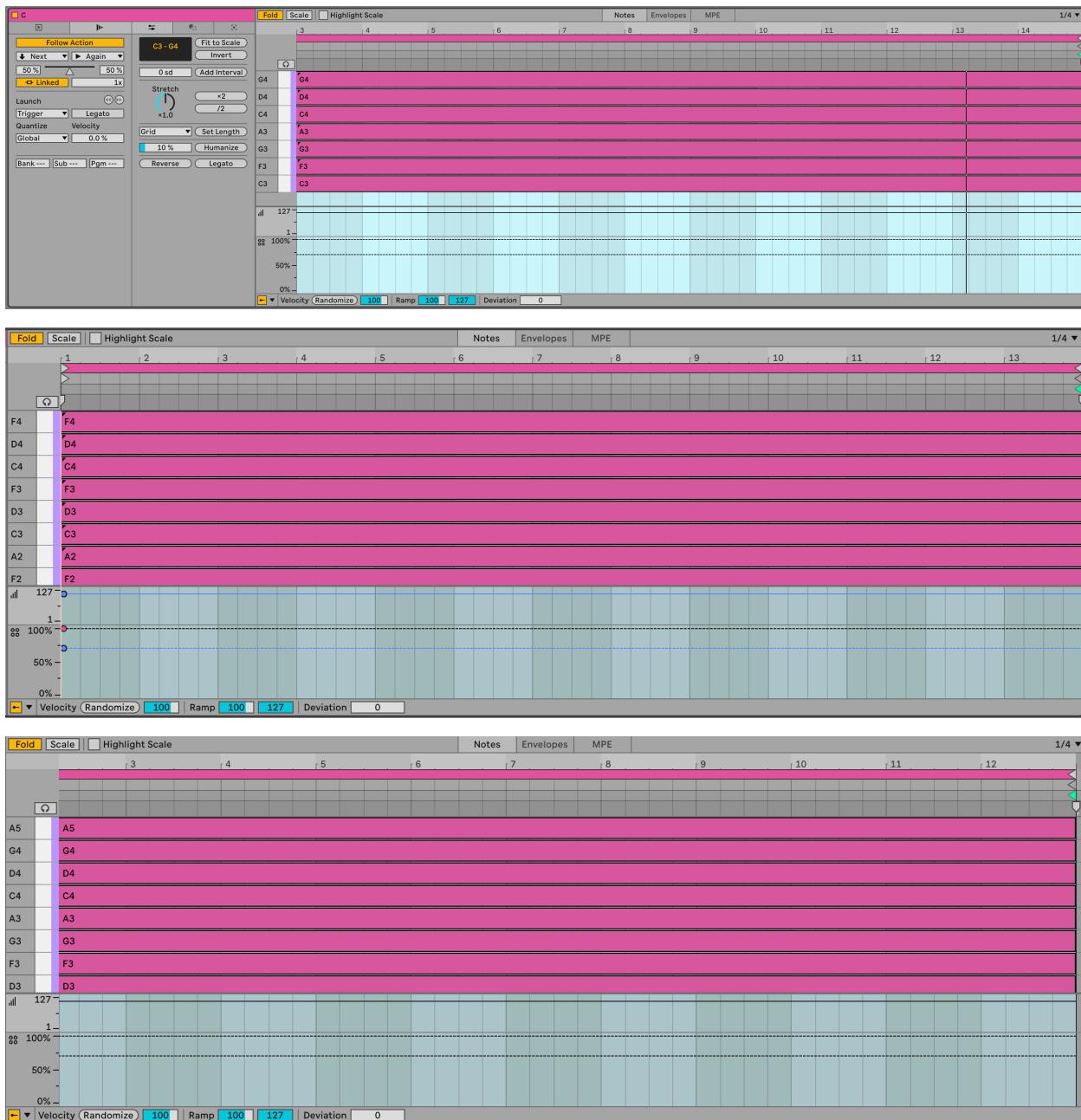
1. Create three MIDI tracks, each with a virtual synth instrument. For this example we're going to use long chords, so I'd recommend a pad or arpeggiated sound. One thing that works quite well is to mix sounds with rhythmic elements, sounds that evolve or where there is modulation to vary the sound over time.
2. Make three MIDI clips in each track.
3. Prepare a MIDI clip with 8 to 10 notes in key (whatever scale you wish) in each clip within the tracks. Avoiding the 3rd of the scale will make the chords tonally "ambiguous", and avoiding (major) 7ths will help the sounds produced blend between tracks and clips (see the chapter on Scales [Process - Using scales](#) for more discussion of scales in generative music). Use the Scale button in the MIDI clip to only show notes within the scale. The chords suggested are arbitrary, but the recipe will work best if you use scales or chord intervals that are "next door" to each other in the cycle of 5ths - [https://en.wikipedia.org/wiki/Circle\\_of\\_fifths](https://en.wikipedia.org/wiki/Circle_of_fifths). See the screenshots below for a visual representation of what I mean.
4. Use random note probability on notes in the chord (see [Process - Note-wise probability](#)). Set the root notes of each clip to 75% probability, and set other notes at about 50-60% probability so that each time the clip triggers, the note choice / chord is different. This will mean that sometimes the chord will have 4 notes, but it could vary between 1 and 7... (in fact it could vary between none and 8, but the extremes are very unlikely). Having higher probability on the root notes favours these notes within the chord, but if that probability is slightly less than 100% then you'll not ALWAYS get that anchoring from the root note.
5. Make sure that the clip lengths vary between tracks (all notes within the clip have the same length). So clip1 in track 1 could have 16 bars length; clip 1 in track 2 could be

## 2 Recipe - Changing chords

13 bars; clip 1 in track 3 could have 10 bars length etc. The important thing is not the clip length, but the fact that they vary **across** tracks.

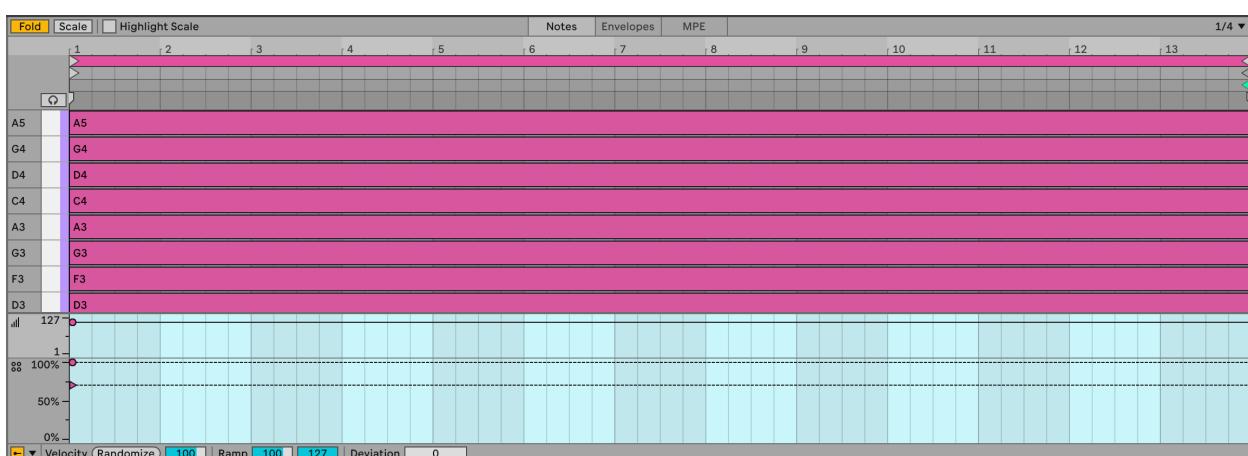
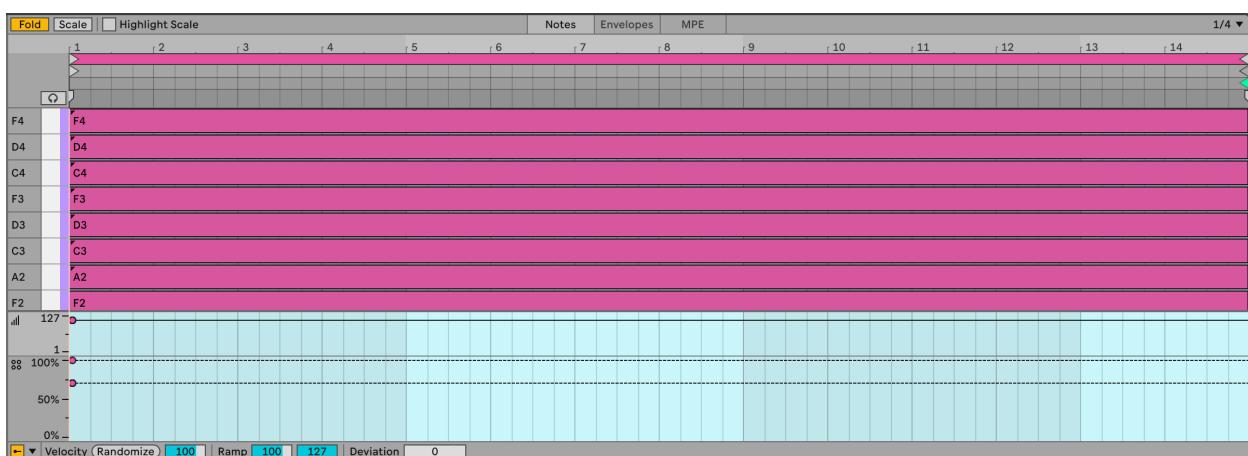
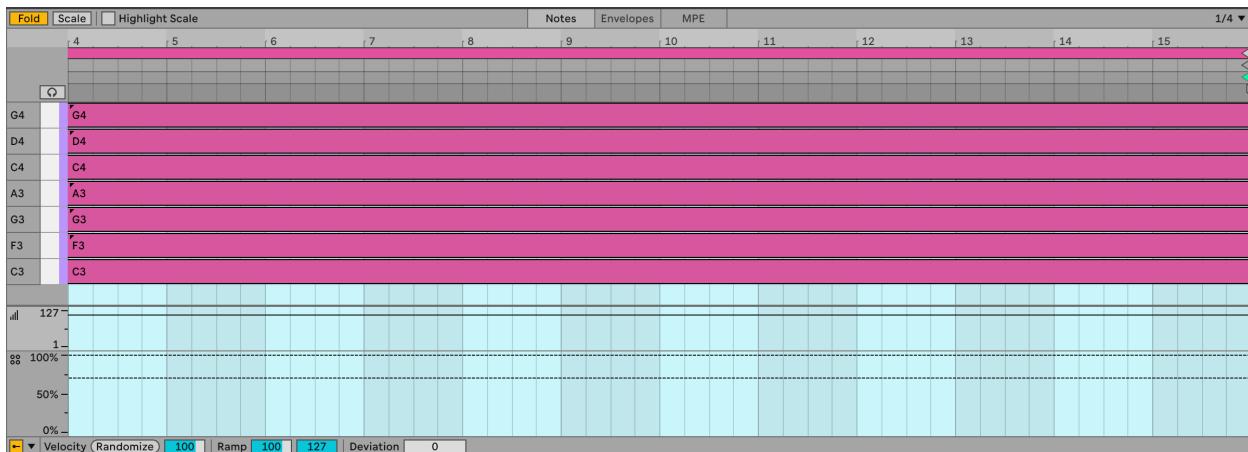
- ALSO, vary clip lengths between scenes **within** tracks. So a track that has 16 bars in clip 1, maybe has 14 bars in clip 2 and 12 bars in clip 3. Similarly, the exact length is unimportant. The important thing is that the clip lengths vary both within tracks and between tracks. AVOID having the same clip lengths in the same scenes. Following the advice above you should wind up with something like this in the first three clip slots of the three tracks:

### Track 1 clips:



## 2 Recipe - Changing chords

### Track 2 clips:



### Track 3 clips:

## 2 Recipe - Changing chords



The clips shown above are *my* choice of notes, length etc. and assume that we're doing a slow moving, textured track. You can play with different choices of notes, shorter lengths of clip, more plucky sounds. Perhaps adding an Arpeggiator or Note Echo to the track would help movement and retain interest (see [Tools - MIDI tools](#) for more details on MIDI effects in Ableton Live).

7. Set up Follow Actions (see the following chapter [Process - Follow Actions](#)) to progress to Next clip at end of each clip with 80% probability, with the other action set to

## *2 Recipe - Changing chords*

“Play again” with 20% probability. When you launch the scene, each clip will play to the end of the clip, but because individual note probability within each of the stacked chords is set to 50-60%, the chord will sound different each time it plays. Also, the randomness we’re using for note probability means that the same chord clip in each track is also likely to sound different. The 80-20% split of next action - “Play again” or “Next” - will mean that some tracks may progress to the next chord while some will play the same clip again.

If the clips are short, then a 50-50% split of next action chance might be more appropriate. In this case because the clips are long we want to force movement onwards. With shorter clips then we might favour repetition so that the listener isn’t bombarded with changes every few beats (see [Process - Balancing the unexpected and the predictable](#) for a discussion of balancing repetition with unexpected elements).

8. Choose sounds for each track that complement each other. You can automate parameters within clips as well to introduce additional movement in the sound. An important point is that the sound shouldn’t be too static.
9. Press play. Sit back and relax.

You can hear one instance of this track here: [https://soundcloud.com/mikeksmith/followactions\\_-chords\\_and\\_arp/s-CKzkRoOEWr](https://soundcloud.com/mikeksmith/followactions_-chords_and_arp/s-CKzkRoOEWr)

(I say “one instance” because this is generative and uses probability. So the next time I render the track it could well sound different. Similar, but different.)

# 3 Process - Note-wise probability

Ableton Live 11 brought us the ability to specify not just the velocity of each note in a MIDI clip, but the probability that that note would play. This is a game changer for the Lazy Producer. Adding variation to MIDI parts is now much easier if you define the lower probability for “fill” notes and higher probability for notes that you want to “anchor” in the clip. Lower probability means that each note is *less* likely to play, while higher probability means that each note is *more* likely to play. A probability of 100% means that the note will **ALWAYS** play.

## 💡 Key idea

Using probability with MIDI notes means that we can create patterns where there are some notes that will definitely play, but some other notes where with each cycle of a loop, there is a chance that the note will play and some chance the note will not play. Using probability moves us away from static and predictable loops into a territory where there is an element of surprise and variation with each loop cycle.

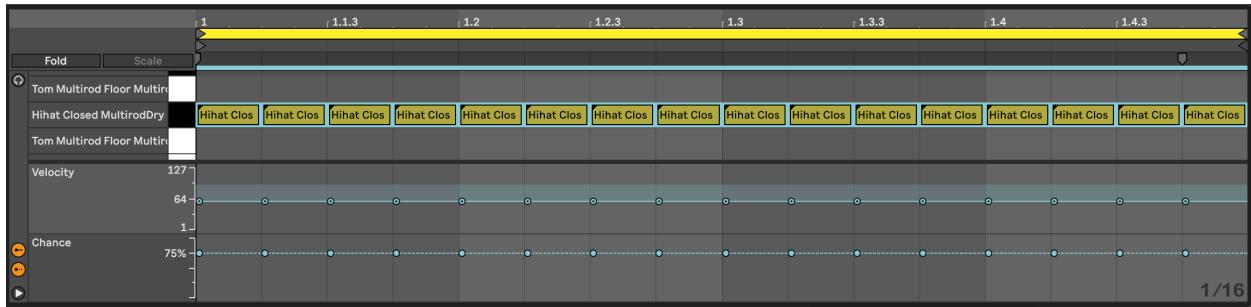
## 3.0.1 Drum parts

One clear use of this is in drum parts. You can program the core drum part that is constant throughout the clip or song, but then add passing notes, “fills” with lower probability. Or reduce the probability on high-hat parts, shakers, claps, lower velocity snare or kick parts so that these only sound once in a while. With this, it’s possible to specify a clip where core elements happen every time, but the “fills” and extra sounds only happen occasionally. Unfortunately, it isn’t possible to specify exactly *when* those notes will or won’t play, just probability, so it’s not possible to specify “Play once, then miss for the next 3 bars” (as is possible with Elektron hardware sequencers).

To see the note chance, click on the MIDI clip then click on the arrow at the bottom of the piano-roll to show velocity. To see the chance / note probability, click on the button below the velocity.

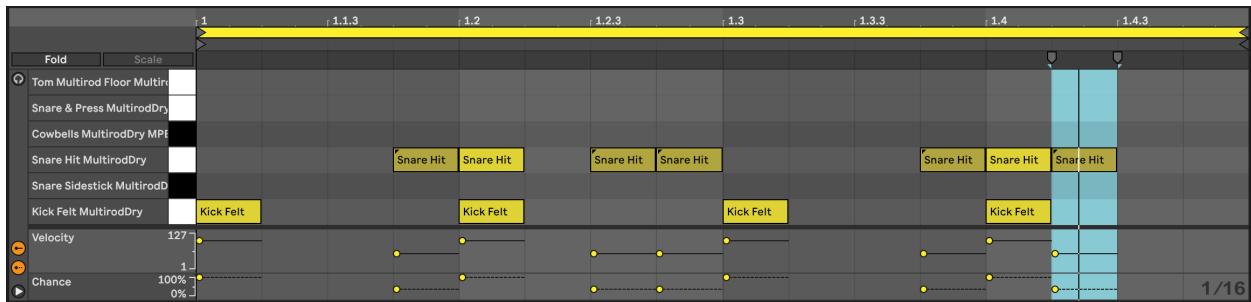
In the clip below we’ve set up a Hi-hat pattern of 16th notes. Note that each of them has a 75% probability of playing, and I have added a little random variation to the velocity of the hits. When we play this clip looped, the Hi-hat pattern will be different each loop and will add a little variation and interest.

### 3 Process - Note-wise probability



Likewise we can add some additional snare hits in the Kick and Snare clip. In the clip below the snare hits on the 2 and 4 of the bar are always going to be present. But the additional hits may only trigger occasionally. These also have lower velocity which gives them the feel of “ghost notes” - barely heard, but present. This will add a little spice and interest to the Kick and Snare parts.

Combined with the Hi-hat part above you get a drum part that has a decent amount of variation without having to program a lot of different clips or parts.

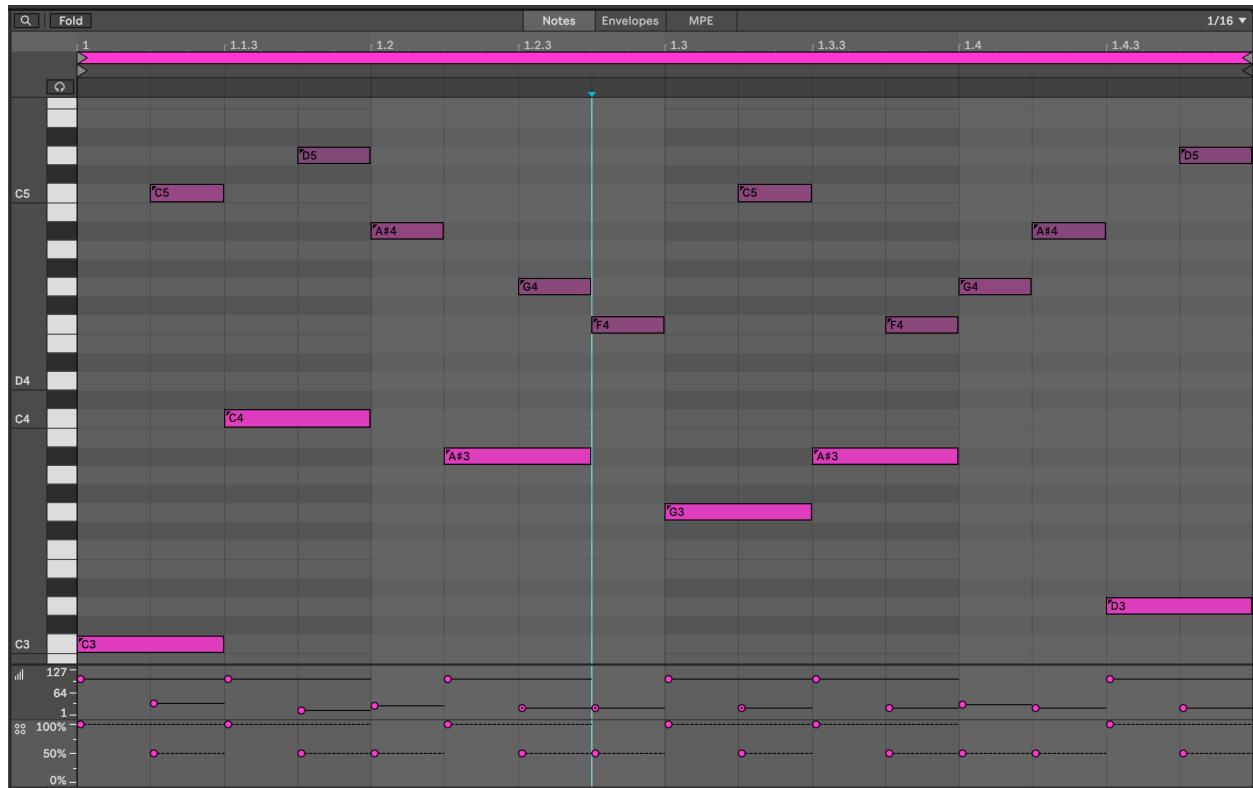


But the note-wise probability extends beyond drum parts.

#### 3.0.2 Lead parts

In the clip below we have a repeating pattern across the notes Bb2-G3. Notes from Bb3-C5 have lower velocity and also lower probability. When we play this clip, we'll have a repeating pattern on the lower notes, but a changing pattern on the higher notes. If you then use the velocity information to change the timbre of the note, then this will add some light and shade into the pattern. This can often be very effective when playing monophonic synth parts, where only one note can play at a time. The higher notes will force lower notes to stop playing which will mean that the lower repeating pattern is never exactly the same each time. Adding a little glide to the sound will also give a little “swoop” as the pitch changes.

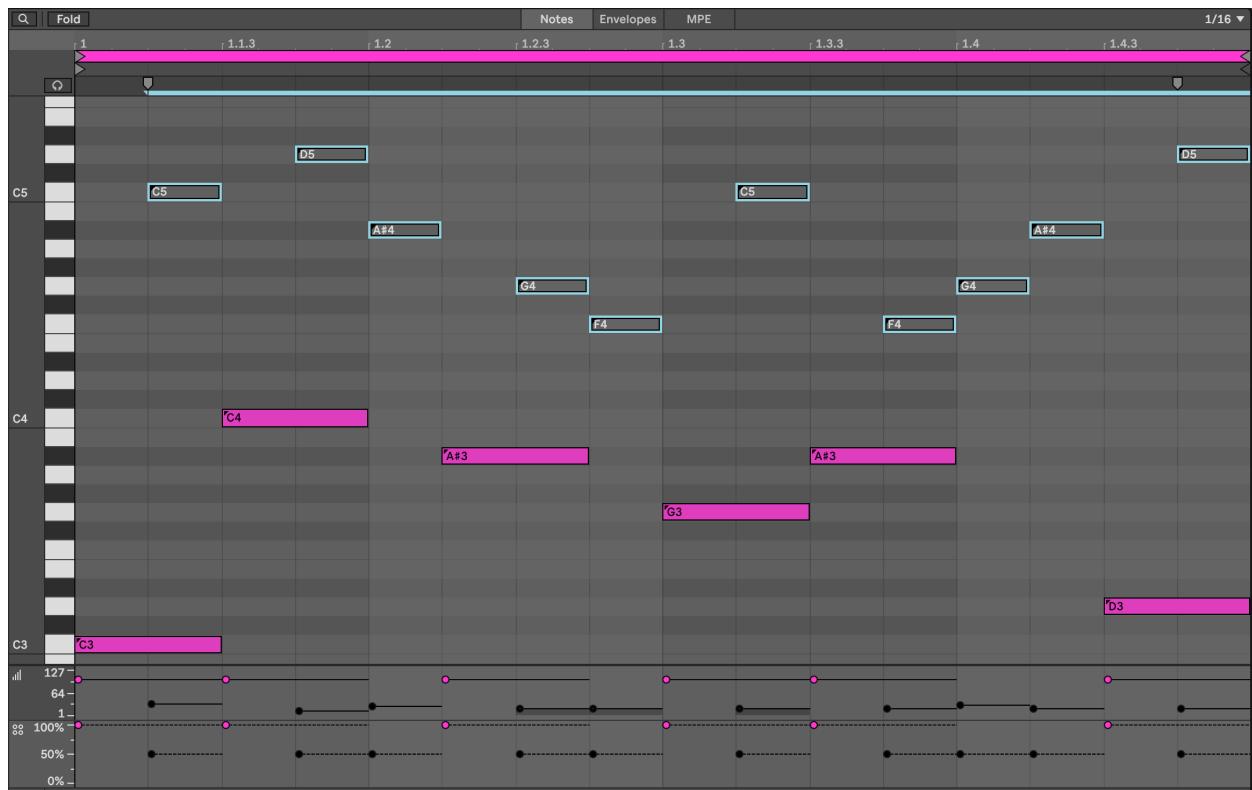
### 3 Process - Note-wise probability



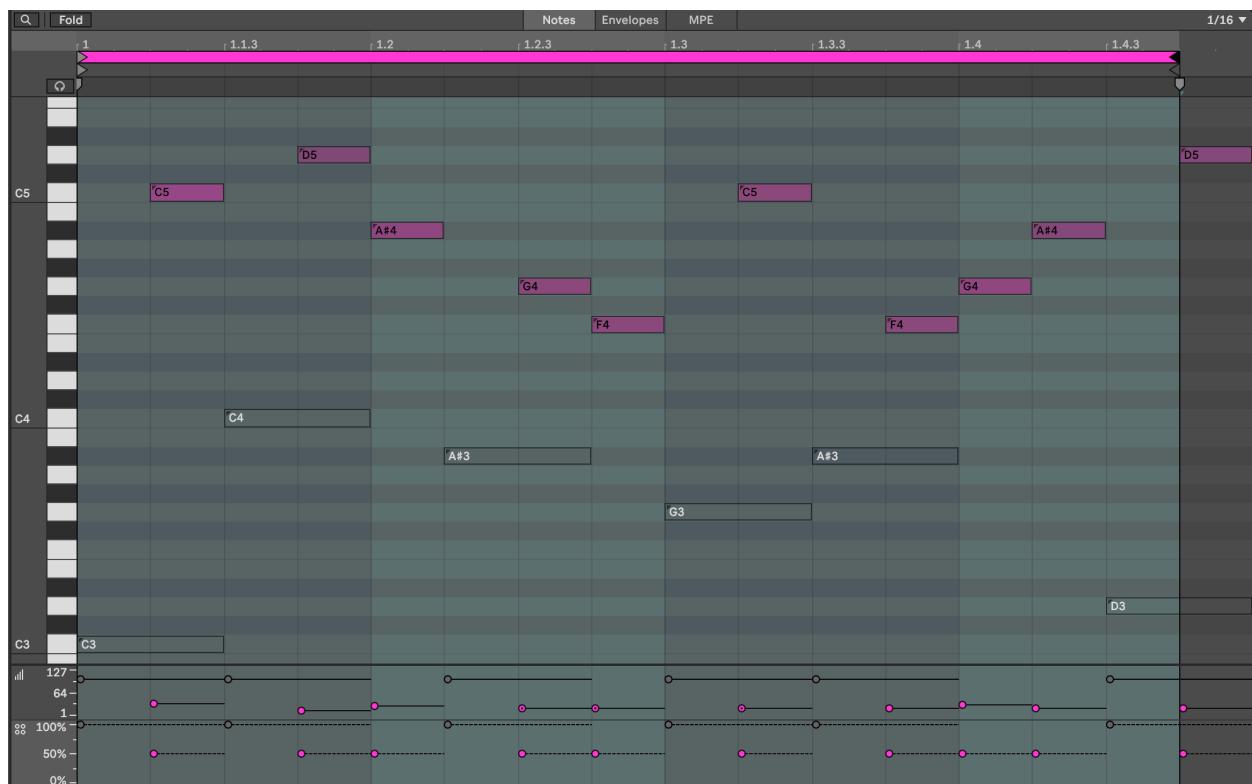
An extension to this technique is to separate the lower, repeating pattern and the higher notes (with lower velocity and lower probability) into separate clips on separate tracks:

Lower part on one track: (Note that “greyed out” notes will not play)

### 3 Process - Note-wise probability



Higher part on a separate track: (Again - “greyed out” notes will not play)



The advantage of splitting the parts into separate clips (and separate tracks) is that you can

### *3 Process - Note-wise probability*

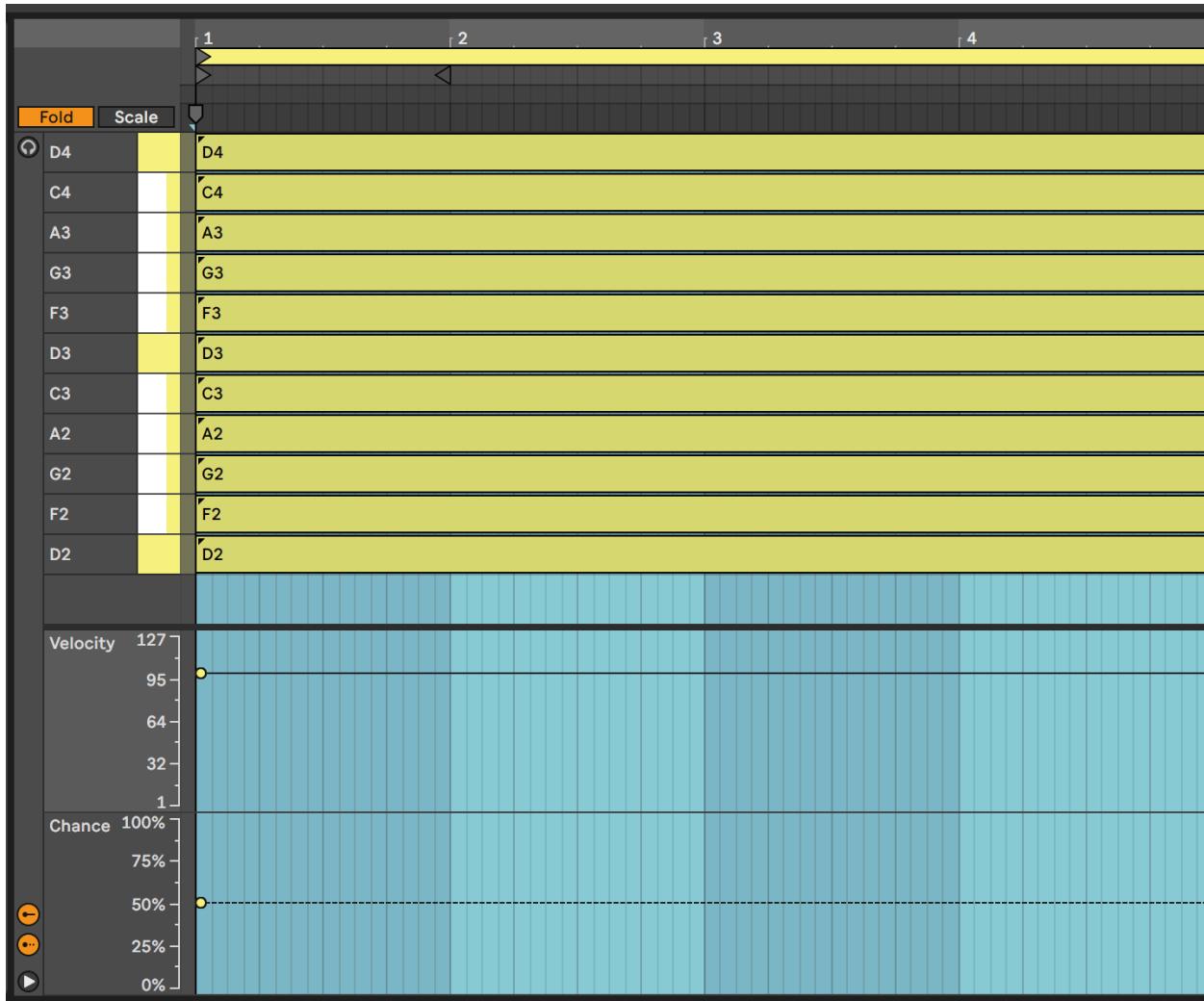
shorten the length of the higher note **clip** - here it's 15 1/16th notes long. That means with each repetition, the notes appear at different positions in the bar and so evolve against the repeating pattern. This is a key concept in balancing predictable and unexpected, as we'll discuss later in [Process - Balancing the unexpected and the predictable](#)

By having the notes in different tracks, you can also keep the repeating pattern "dry" and apply delay or echo to the upper part. You could also opt to modulate or vary the timbre and sound of either track to keep the repeating part interesting.

### **3.0.3 Chord parts**

In [Recipe - Changing chords](#), I suggested using probability on the notes within chords. If you add 8 MIDI notes in a chord and specify 50% probability for all of them, then Ableton is going to (effectively) toss a coin for each note. In the real world, when we toss a coin 8 times, we don't always get 4 heads and 4 tails. Sometimes we get 3 heads, 5 tails (or vice versa), and occasionally outcomes like 2 heads, 6 tails. Imagine if in the coin toss, heads means that Ableton plays the note. Then the number of notes that gets played will vary each time the chord is played - sometimes it will be 3, sometimes 6 etc. This means that you can specify as many notes as you like, and tune the probability until you get something that you like the sound of, depending on whether you want more or fewer notes in the resulting chord. It also means that careful choice of notes from the scale within the chord allows you to sometimes get rich, complex chords, sometimes chord inversions.

### 3 Process - Note-wise probability



By specifying higher probability for “root notes” in the chord, you can anchor the chord to a particular scale. As a Lazy (but creative) Producer, you can also allow the tonality (which musical scale) is being implied to vary so that the chords that emerge could fit with a variety of bass root notes, perhaps supplied separately in a different track. Avoiding the third (minor / major) or seventh of the scale can make it easier to achieve this kind of fluid and ambiguous tonality. For more discussion on scales and note choices see [Process - Using scales](#)

In that same recipe, I have applied this idea across tracks as well. Using the same MIDI chord idea across all tracks, but then changing clip lengths within tracks (so each clip within a track has a different length) and between tracks (so that adjacent clips in a scene have different lengths) and then coupling this with Follow Actions as described in [Process - Follow Actions](#) to progress through the clips, means that the chord patterns change more subtly and gradually than having block changes in tonality and scale every 4 or 8 bars. Including note-wise probability within clips also means that the next time the clip plays you get a slightly different result. Again, balancing something predictable - chords with a sequence of tonality / predictable root notes - with the unpredictable - exactly what notes are being

### *3 Process - Note-wise probability*

played from the chord and that occasionally the chords could work in both keys - means that the listener takes longer to get bored with the sequence.

#### **3.0.4 Live 12 - Grouped notes**

Ableton Live 12 brought the ability to group MIDI notes and then define whether *all* of the notes in that group will play or only *one* note within the group. This can apply to notes in a chord, or notes in a sequence - basically any of the notes that you define to be within the group. This can be useful as it means that within the group you can define that *exactly one* of the notes will play, or that *all* of the grouped notes will play, which is slightly more deterministic (rules based) than the probability approaches above. Having said that, you can also apply probability to the group, so that with say 50% probability, all of the grouped notes will play - allowing you to write a specific set of “fill” notes in a drum part that will play with a given probability. You can also set probability along with the “play one” group rule. We’ll look at some concrete examples in [Tools - Live12 Overview](#).

# 4 Process - Follow Actions

Ableton has had “Follow Actions” within Session View clips for a *long* time. Follow Actions allow you to perform a variety of actions at the end of a clip or after a given number of bars and beats. Follow actions are generally used where there are a number of clips within a track, allow the performer or Lazy Producer to automate moving from one clip to another. This allows you to determine a sequence of events - moving from one clip to another or playing the clip again, etc. There are a number of really useful uses for follow actions for live performance, but it also has incredible uses for making generative music. Follow Actions help govern the amount of variability in our generative sequences - whether there is a definite sequence of events that happen predictably, or whether there is an element of chance introduced.

## Key idea

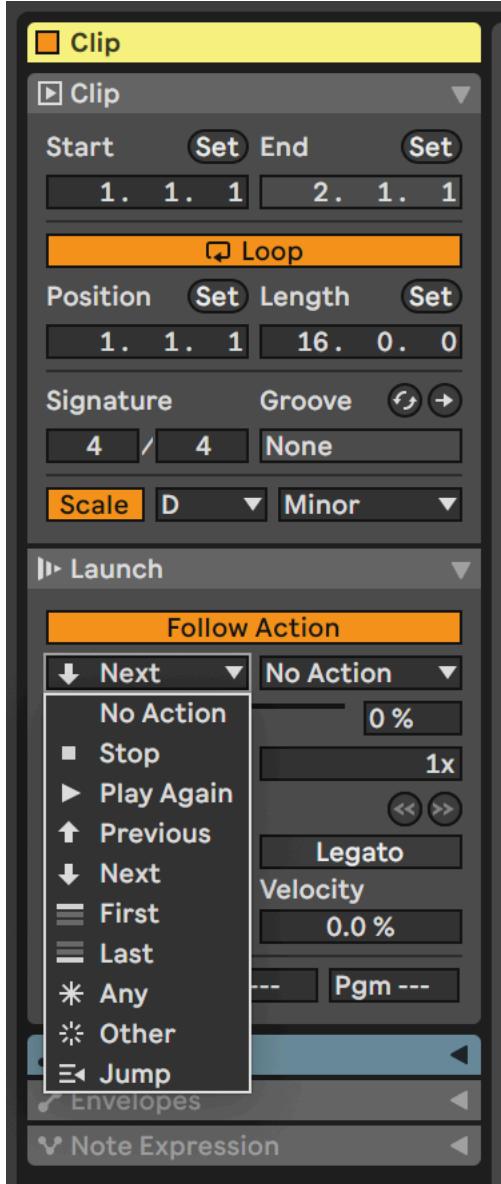
Follow Actions allow the Ableton user to automate moving from one clip in a track to another. This automation can happen at the end of the clip or after a given number of bars and beats. In order to use Follow Actions, the clips in the track have to be contiguous - that is, no gaps between the set of clips.

Even if the notes inside a (MIDI or Audio) clip are fixed, Follow Actions can be used to move from one clip to another. This allows the Lazy Producer (or Lazy Performer!) to set up clips that contain pre-composed riffs, rhythmic patterns, melodic lines, chords etc. and then get Ableton to help sequence these in interesting ways.

Clips can also contain nothing at all. And these empty clips can have different lengths. So if you need space between melodic phrases, you can get Follow Actions to play an empty clip. At the end of the empty clip, you could play more of nothing, or jump to a specific clip that holds the next phrase.

## 4.1 Follow ACTIONS

To see the Follow Actions for a clip, click on the “Launch” item to expand the section. To enable Follow Actions for the clip, click on the button marked “Follow Action”, and ensure that it is active / orange.

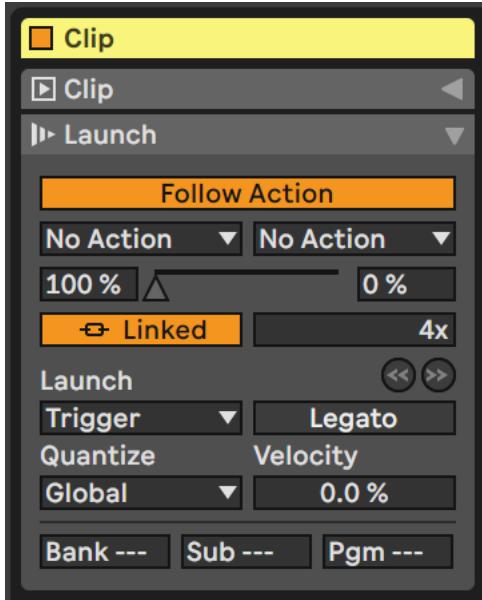


There are many different possible actions with the Follow Action. These are mostly self-explanatory but there are a few subtleties that are worth pointing out. As we point out above, Ableton needs to have contiguous sets of clips within a track to enable Follow Actions. The “First” and “Last” options above refer to ***the first and last clips within that contiguous set***, not necessarily the first and last within the track.

The “Any” option randomly jumps to any of the clips within the contiguous set, including the current clip. “Other” randomly jumps to a different clip within the set. “Any” can be useful when you want to allow a repeat of the currently playing clip, while “Other” is more useful to force a jump to a different clip. “Jump” allows you to nominate a specific clip to jump to. “No Action” can be used as a “catch all” default action. With “No Action” in the alternative (right hand) Action, you can automate Ableton moving to a different clip after a certain number of bars and beats, but if this option is not taken then the clip continues playing to the end of the clip.

## 4.2 FOLLOW Actions

In previous versions of Ableton Live, Follow Actions were set up so that the user had to specify a number of bars, beats and subdivisions before the action would happen. This leads to a lot of mental arithmetic to determine how many bars and beats AFTER the clip starts to make the jump. Fortunately now from Live 11 the default is to apply the Follow Action at the end of the clip, as long as the “Linked” button is active.



There is also the option to play the clip a number of times before applying the Follow Action. This helps us balance between the familiar and newness - hearing something repeated helps reduce the randomness and uncertainty of constantly introducing new things. While repeating one part or track you can introduce something new with a different part or track.

The Legato option is also useful. If the Follow Action is applied in one clip, and if Legato is active, then the clip that is triggered next will pick up from the number of bars and beats in the previous clip (where Follow Action was applied). This can be useful for applying different endings to clips (the equivalent to fills in drum / rhythmic parts). That is, you can play 3 bars of one clip, then use the fourth bar of some other clip to provide the ending of that clip, then apply a Follow Action to go back to the first clip, on to a different clip or loop around to the beginning of the currently playing clip. This allows us to set up automation that introduces musical ideas, rather than changing clips together, or only at the end of clips.

## 4.3 Follow Action probabilities

By default Live will perform the specified follow action with 100% probability i.e. it's guaranteed to happen. But there is a second option where the action is performed with a given

probability, and with 100% minus that probability it will perform a different action (probabilities MUST add to 100%). For example, it could do action 1 with 70% probability and action 2 with 30% probability. Like the coin-flipping in Note-wise probability that we talked about in the previous chapter [Process - Note-wise probability](#), here we have a **biased** coin, determining what happens next. With 50% - 50% Ableton flips the coin and chooses which action to take next. With 70% - 30% the left-hand action will happen more, but there's still a chance of the right-hand action happening.

The automation and probability here allows a HUGE range of possible journeys through a sequence of clips. If you introduce a 50% Play Next probability on the clips in the Changing Chords recipe, then it will stretch out the changes across clips as some of them will loop and continue playing the same chord for longer.

## 4.4 A Random Walk

In statistics there is a thing called a “[Random Walk](#)” where we take a step forward or backward with a given probability, or stay where we are. With certain probability set ups, it’s possible to stay in the system for a LONG period of time, taking steps.

Imagine that you go for a walk from your front door, but toss a coin before every step. Heads you move two steps forward, tails you move one step back. How long before you walk a mile?

Random Walks can also have “absorbing states” at the end. This would be an equivalent of a “Stop” action in the Follow Action dialogue box above. Using random sampling and probabilities you could simulate and work out the likely playing time of your generative piece.

# 5 Tools - MIDI tools: chords and added notes

There are many [MIDI tools and devices](#) within Ableton Live that will help manipulate MIDI notes - shifting pitch, adding random pitch deviation, applying arpeggiation (cycling through notes in a chord), adding additional notes to an incoming note to make a chord, changing the length of the played MIDI note, as well as manipulating incoming MIDI control messages - using the modulation wheel to control expression, changing velocity, modulating other devices based on velocity, MPE and other control messages.

## 💡 Key idea

Adding notes, randomly changing the pitch of notes, changing the lengths of notes and generating chords within a given scale using MIDI plugin tools can form the basis of a generative piece. Using other MIDI plugins to arpeggiate these chords and add additional notes and harmonies will produce a constantly evolving sequence of MIDI notes within the scale.

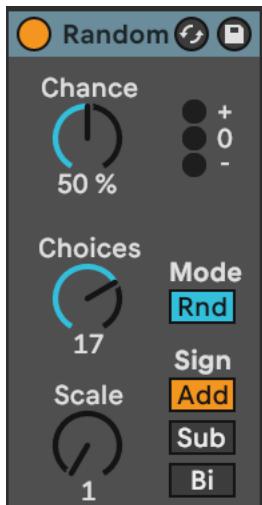
## 5.1 Ableton's Note Echo plugin

[Ableton's Note Echo](#) plugin may at first seem a bit pointless. I mean you can take the output from an audio track and just apply a Delay or Echo to that, so why bother doing it on the MIDI side? Well, because with Note Echo it will adjust the velocity of each echoed note, and you can also pitch up or down the echos, leading to changing pitch. You can also combine Note Echo plugins to produce nice “multi-tap” delays, each with different characteristics. The echos can build up quickly if you stack these, so I tend to use them with minimal or no feedback. Combined with the Random plugin discussed below you can easily build up evolving and interesting patterns which are similar without repeating. The “Feed Delay” amount controls the amount that the velocity decays with each repeat.



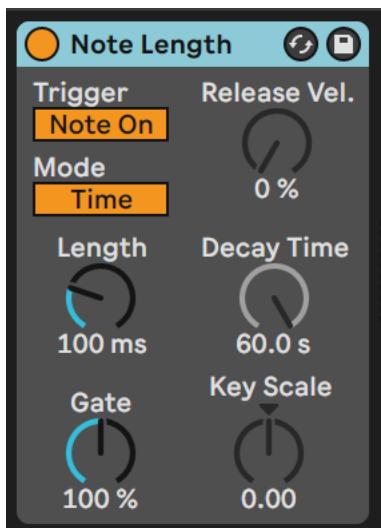
## 5.2 Ableton's Random plugin

The [Random plugin](#) changes the pitch of the incoming note within a range specified within the plugin. In the example shown below, incoming notes will be randomised to an increase in pitch up to 17 semi-tones above the original note with 50% probability (flip of a coin). The “Sign” switches allow you to define whether the random pitch change is added, subtracted or either to the incoming note. The “Mode” switch allows you to swap between randomness and incrementing, if you want the pitch to change gradually with each incoming note. The latter might be useful if using this with a sampler where different samples are associated with different MIDI notes. This would allow you to cycle through samples using this device.



## 5.3 Ableton's Note Length plugin

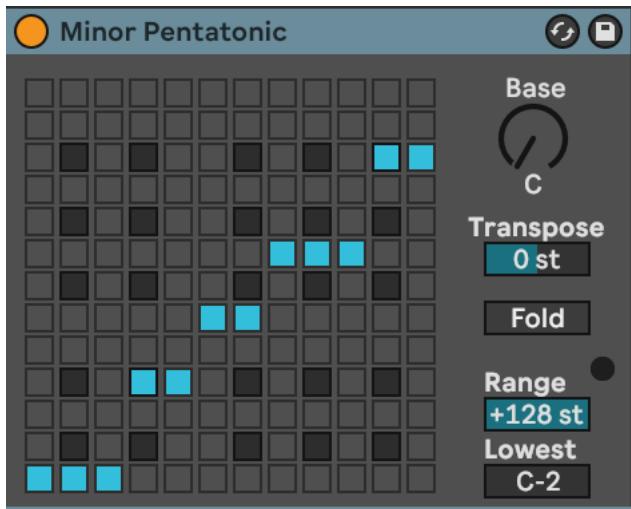
[Note Length](#) is a really useful plugin to help modify the incoming MIDI and make notes longer or shorter. The “Gate” control can be used to double an incoming notes length, while the “Length” control defines the length of the resulting MIDI note in milliseconds or seconds. I use this to take incoming MIDI information and “smear” it for use with pad sounds. This is useful in generative music, because you can relate the pad chord using these smeared notes to any incoming moving MIDI part. Be aware though that long “Length” settings, it’s possible to stack up MANY midi voices in a plugin. Limiting the number of voices in the plugin instrument or reducing the note length can prevent CPU overload!



## 5.4 Ableton's Scale plugin

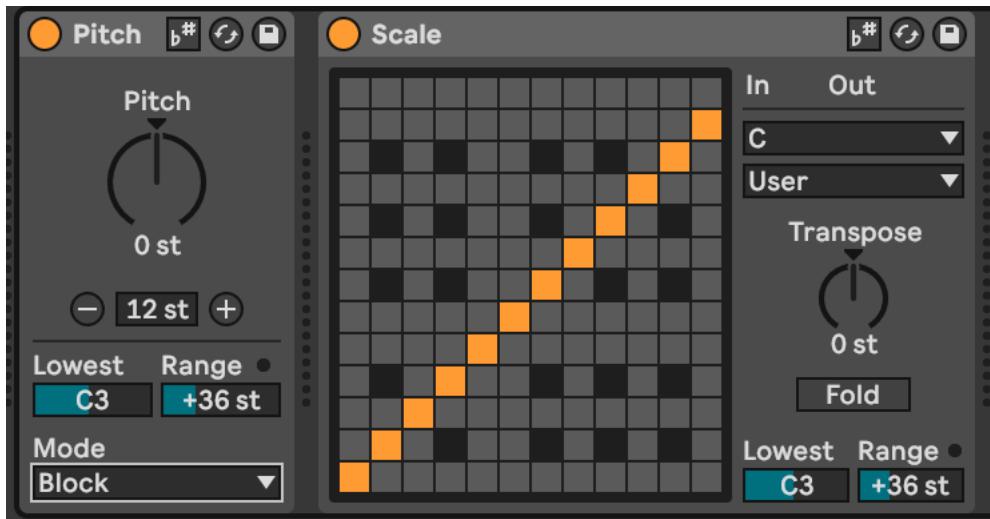
The [Scale plugin](#) is the generative musician's friend. It takes whatever MIDI nonsense you throw at it and quantises it to a given scale. ***This isn't cheating.*** Well, it kind of is, but it's a well used cheat within the modular synthesis world where quantisers have been used for a long time to ensure that control voltage (CV) signals conform to a particular scale.

## 5 Tools - MIDI tools: chords and added notes



### Note Range

The MIDI Pitch and Scale devices have the option to limit the range of notes played. You can specify the Lowest MIDI note (default C-2) and the Range (default 127st) but you can also use these to limit the MIDI note range for a part and what to do with MIDI notes that fall outside the range (Block not to play, Fold to have the note folded back into the nominated range or Limit to play the upper end of the note range specified). When generating parts using techniques here, it's sometimes useful to ensure that you don't have unexpectedly very high or very low notes within a part.

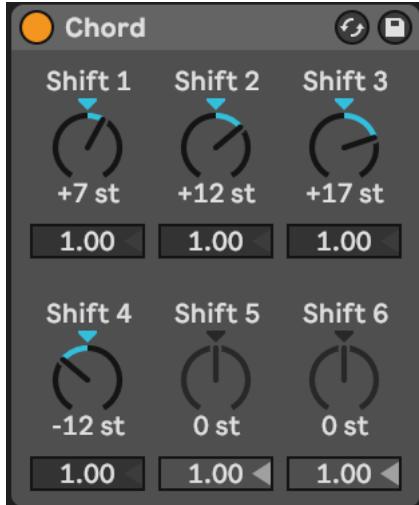


## 5.5 Ableton's Chord plugin

Ableton's [Chord plugin](#) adds notes to incoming MIDI notes to form chords. Up to six additional tones can be added to the incoming MIDI. Each tone has its own associated

## 5 Tools - MIDI tools: chords and added notes

velocity which can help accentuate certain parts of the chord over others. The Chord plugin adds notes to **every** MIDI note it receives, and in fixed intervals above or below that incoming MIDI note. So again, it may be a good idea to use the Scale plugin to ensure that the generated notes and chords fall within a desired scale.



Coupling the Chord plugin with the Arpeggiator and Random plugins is a good way to make arpeggiated lines that do not repeat exactly. As with most of these MIDI plugins the order you specify them can provide interesting variations. So Chord -> Arpeggiator -> Random -> Scale adds random notes into the arpeggio resulting in cascades of notes that change note to note, but within a constrained set of output notes. Random -> Chord -> Arpeggio -> Scale would lead to a consistent arpeggio for the duration of the input note but could be different for each cycle of the generating MIDI loop.



## 5.6 Probability Gate

The [MightM4L \(MML\) Probability Gate](#) is a Max for Live MIDI device which allows you to set a probability (in %) and it then allows that proportion of notes through the device. This can be a very useful device to use with “busy” MIDI generators like the Turing Machine (see [Tools - MIDI Generators](#)) which are constantly generating 1/8th or 1/16th notes. Use of a probability gate “thins out” that stream of output.



## 5.7 Deviate

The free [Deviate Max for Live plugin from Novel Music](#) can add variation to incoming MIDI notes. The “TRIGGER” section defines how often the deviation or variation will occur - so it is possible to balance the unexpected with the predictable, or keep things the same before introducing randomness. One important aspect of Deviate is that it has a memory, so if you like the random pattern generated, you can “lock” this by moving the “LOCK” slider to the right. This means it acts a bit like the Turing Machine devices used in modular synthesis. Deviate has two sides - MIDI, which we’ll cover here and MOD which we’ll cover in the section .

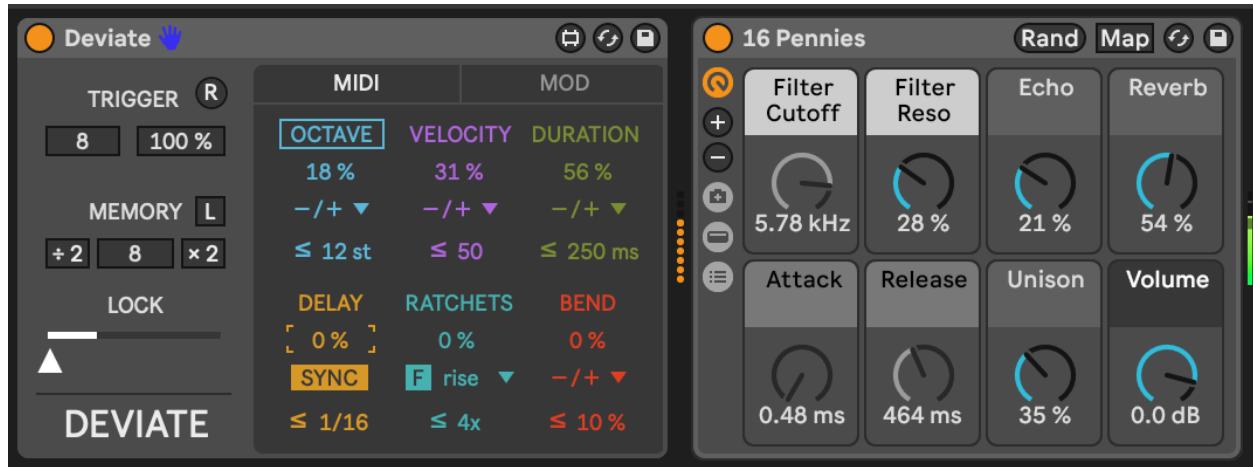
The “TRIGGER” section is worth spending a little time to understand. It determines how many MIDI notes will pass before the deviation is triggered. So if set to “1” then every note triggers a new deviation. If set to 8 then every 8 notes the deviation settings may be triggered, but then these deviation settings persist for the next 7 notes. The probability to the right (below set to 100%) means that the deviation will be triggered every time, but if lower than 100% then that probability determines whether *any* deviation will occur. So if you set 8 notes, with 100% probability then the Octave deviation may trigger 18% of the time, but if it does then notes will sound either an octave up (or down) for the next 7 notes.

The “MEMORY” section works along side the “LOCK” switch. Deviate can “listen” to a sequence of MIDI notes of the nominated length (here 8 notes) and will keep track of what deviations were applied in that 8 note sequence. If you like what you’re hearing you can “LOCK” the deviation settings in by sliding the “LOCK” slider to the right. As you bring the “LOCK” slider back to the left it will start to reintroduce new deviations while retaining the memory of the locked settings. With the “LOCK” slider all the way to the left, the “TRIGGER” and Deviate settings are applied according to the “TRIGGER” settings. Deviate can also learn how many notes are in a MIDI clip if you click on the “L” button

## 5 Tools - MIDI tools: chords and added notes

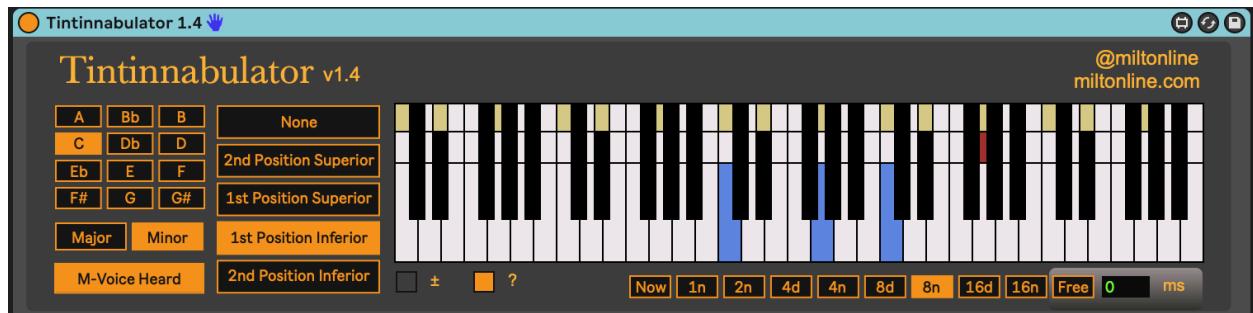
next to “MEMORY”. The memory sequence can be divided by 2 or multiplied by 2 using the adjacent buttons.

It is worth reviewing [the YouTube tutorial for this device](#) as it can do *a lot*.



## 5.8 Max for Live Tintinnabulator plugin

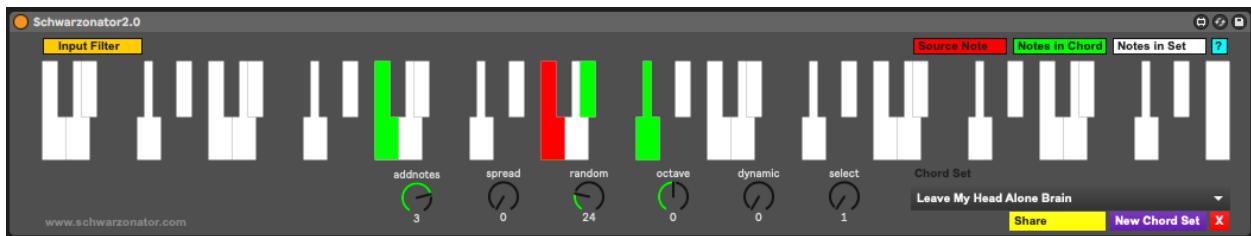
The [Tintinnabulator](#) Max for Live plugin by Milton Mermikedes is an implementation of Arvo Part’s [Tintinnabulation technique](#). This technique arpeggiates the original chord, and then adds additional notes / harmonies to those notes based on the nearest tone from the scale triad. So 1st position inferior C minor takes a C minor triad (C - Eb - G) and assigns a harmony to the incoming notes using the nearest chord tone 1 position lower than the incoming note (the M-Voice). Within the plugin you can specify the delay between the original notes and the additional harmonies. The good news for producers of generative music is that the choice of T-Voice position (1st Superior, 2nd Superior, 1st Inferior, 2nd Inferior) can be set to be random. This will add variety and keeps the T-Voice harmony changing. This plugin can be used on melody lines or chord inputs for a variety of different effects.



## 5.9 Max for Live Schwartzonator plugin

The Schwartzonator plugin (<https://www.ableton.com/en/packs/schwarzonator/>) combines chord generation with scale quantising. You specify what notes are “valid” for the output - quantising input to output - and then the knobs below the “piano” define the makeup of the output chord. The “add notes” knob defines how many additional notes you want to add to the input MIDI note to create a chord. “spread” spreads these out over the keyboard. “random” changes the disposition of the chord - creating chord inversions and changing the output chord when it’s triggered by the input MIDI note. With “random” increased you can get ever-changing output chords for the same input notes. “dynamic” adds random velocity across the output notes. “select” allows you to automate selection of the Chord Set.

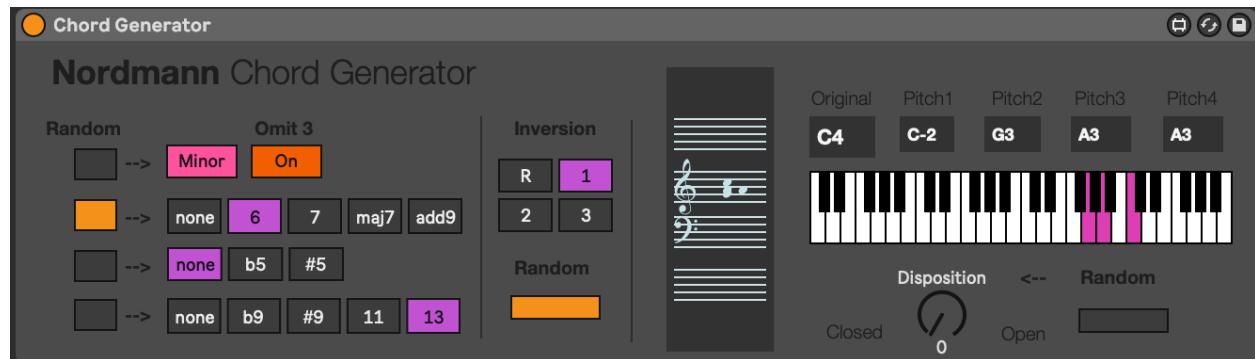
Schwartzonator was designed to help musicians who did not know musical theory generate chords from single input notes and the scale or chord quantising allows the user to generate musical ideas that sound good. For the generative musician there are many things in this plugin that are appealing - the ability to turn single input notes into chords that conform to a scale or chord pattern and the ability to also add randomness to ensure that the chords are different each time they are triggered. These coupled with the MIDI tools above can help create evolving patterns that fit with an overall scale key, chord pattern etc. defined for the track.



## 5.10 Max for Live Chord Generator plugin

The [Nordmann Chord Generator](#) Max for Live plugin is similar to the Schwartzonator above in that it takes input MIDI notes and turns them into chords. The difference is that while the Schwartzonator works by defining notes that can be used or not in the output, the Nordmann Chord Generator assumes a little more musical theory that the user knows what adding a 6, 7, add9 means. Of course, you can listen and find out... Also not every note in a MIDI scale is going to produce output chords that are in scale if you use a “b5” chord with them. In this way, using the MIDI scale plugin after the Chord Generator may help quantise the MIDI to sound reasonable in the context of the scale, although selecting to use a “b5” chord and quantising to perfect 5th may seem counterintuitive.

## 5 Tools - MIDI tools: chords and added notes



# 6 Process - Routing MIDI and Audio

In the spirit of “The Lazy Producer” I like to reuse MIDI and Audio inputs across many tracks, often adding MIDI effects (as discussed) or audio effects. The benefit of this is that the tracks which reuse the same input are related to the original in pitch but with different timbre if they use different synth sounds, or have different audio effects applied. In Ableton Live it’s trivial to route one track (either MIDI or audio) into another - either through “pushing” the output of a track to another or specifying that a track takes the output of another as “input” and then monitoring that input.

## 💡 Key idea

Reusing MIDI or audio tracks across many channels ensures that the different tracks are related in their input source, even if additional MIDI or audio effects are applied. This provides a level of coherence between sounds and tracks, the “predictable” in our balance between unexpected and predictable. This helps the listener hear the sounds as coming from the same “family” even if the actual sounds in each track are different. Exploiting this, it’s possible to layer up tracks that are pretty much guaranteed to work well together, while exploiting different sound sources, timbres, even pitches (if using MIDI input) or resonances (if using Audio input).

In Ableton Live we have a couple of ways to either “push” or “pull” MIDI or audio from one track into another. In the image below, MIDI received by Track 1 is “pushed” (via MIDI To choices) to Track 2 which contains an instance of Operator. The second box below the MIDI Track specifies that the MIDI is to be sent to the input of the first MIDI device in this track - the Operator instrument in this case. You don’t **have** to have this specified, but in this case it means that the Monitor setting in Track 2 can be left on “Auto”. You can also route the input to a specific device in an instrument rack. You can use this approach to send MIDI from multiple MIDI tracks to **the same device**. For example if you have separate MIDI patterns for Kick, Snare and High-hat these can be sent to the same instrument track containing a Drum Rack. This can be useful if you want to apply MIDI effects, randomness, follow-actions or modulation to the high-hat but not the kick.

The third track (MIDI) here is “pulling” the MIDI from Track 1. Note that the Monitor is set to “In” so that it “listens” for the MIDI being sent. In this case I can add a MIDI device like Note Length to “stretch” the incoming MIDI and then apply an Arpeggiator to that to produce new moving parts that are related to the sequence in Track 1, but different.

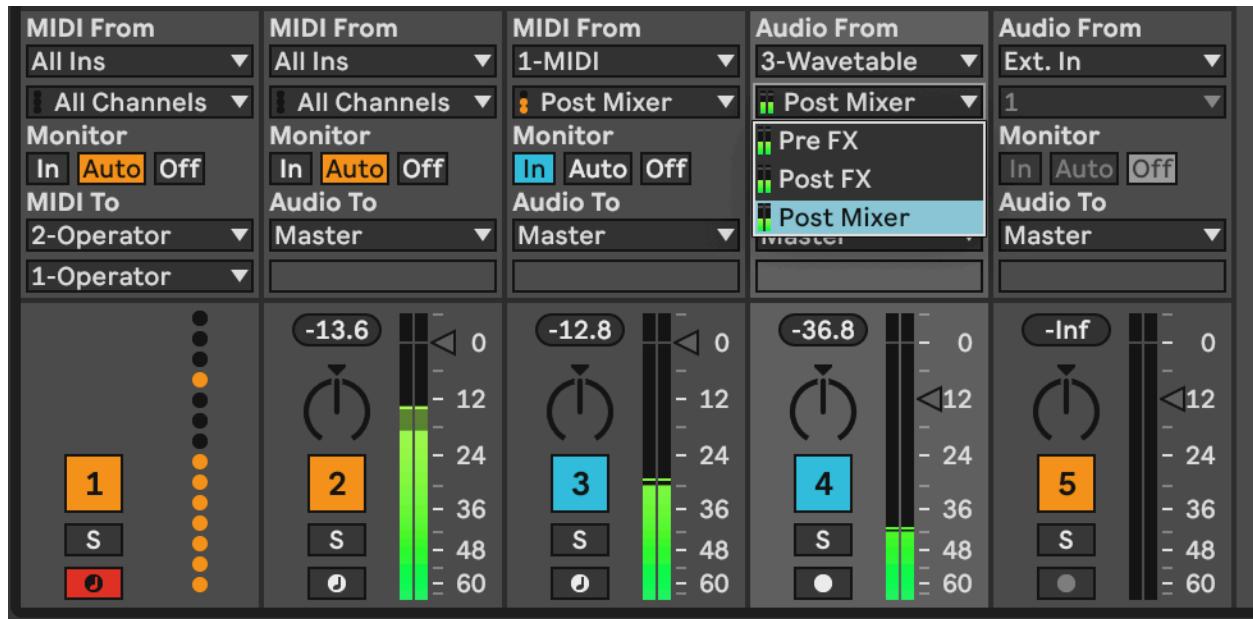
The fourth track (Audio) here is “pulling” the audio from Track 3 (Wavetable). Similar to Track 3, it needs the Monitor set to “In” to “listen” for the audio output. In this track we’re

## 6 Process - Routing MIDI and Audio

also showing the various options for where to grab audio - it can be “Pre FX”, “Post FX” or “Post Mixer”. “Pre FX” means that the audio is grabbed by the track before it goes through any audio processing on the input track. This means you can apply different effects to the same raw audio source - some on Track 3 and some via Track 4. “Post FX” means that it grabs audio after effects have been applied. “Post Mixer” means that track volumes and panning are applied prior to grabbing the audio. “Post Mixer” might be handy if you wish to control levels from only one track.

In MIDI tracks the “Pre FX” and “Post FX” apply to MIDI effects as well, so you can grab MIDI inputs either before MIDI effects are applied, or afterwards. Since Ableton’s track routing works this way consistently, it’s possible to add intermediate MIDI tracks with no instruments, but “listening” to the MIDI output at a particular state (“Pre” or “Post” FX on the source track) and then have additional tracks that exploit the intermediate track. This flexibility and ease of routing is a massive bonus for The Lazy Producer. It’s possible to reuse inputs across multiple tracks in multiple ways to do a huge range with them. In particular I use this technique to build texture, especially in “drone” tracks where I can apply different resonators, textural effects to audio inputs from the same source and then blend these across tracks.

An alternative approach to the same problem would be to employ “Instrument Racks” or “Effect Racks” in one track. However, the benefit from the approach above is that each track can then use Return track sends to apply effects. It also makes live performance somewhat easier since you can quickly see the levels of each track and control these via a MIDI controller device like Ableton Push.



## **6.1 Splitting MIDI for a single instrument across tracks**

As we discussed in [Process - Note-wise probability](#) it's possible to use multiple MIDI tracks that send MIDI to a single instrument. This might be useful to split repeating and random parts, so that you can apply MIDI effects to the "random" part while keeping the repeating part static.

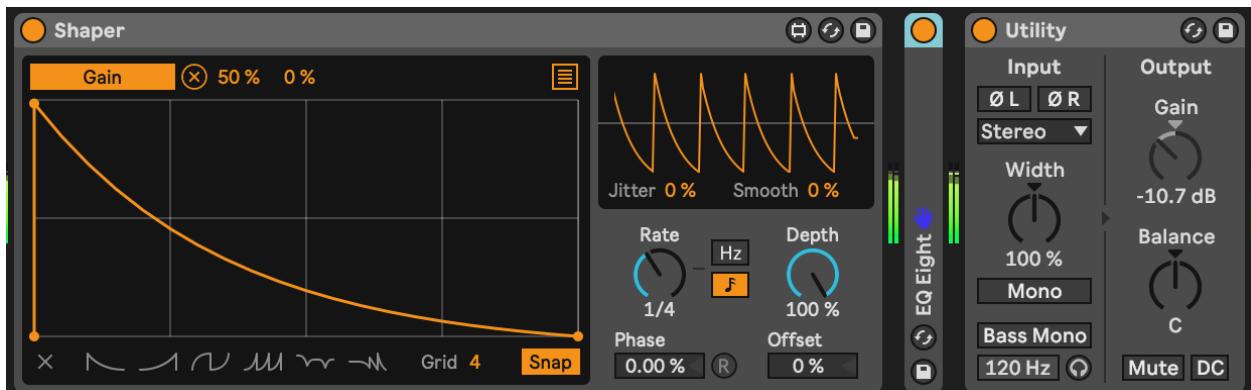
A similar technique might be used to have distinct MIDI parts for a Drum Rack, so you can keep certain parts "locked in" and repeating, while other parts have additional MIDI processing and add flavour to the part. Even within that framework you could have a repeating kick and snare part, but add additional snare fills, flams or lower velocity random hits to add variation and spice to the snare part.

The flexibility and ease of routing audio and MIDI around in Live makes it a very versatile platform for creating complex signal paths, and apply different effects and modulation at a variety of points both before the MIDI information is used by an instrument, and after the audio is created via the instrument and effects.

## 7 Recipe - Doing more with less using MIDI tools and plugins

In this recipe we'll use the simplest MIDI generator, in this case a single MIDI note, with MIDI effects and routing to create a track - these are covered in [Tools - MIDI tools](#). This recipe also uses MIDI and audio routing in Live quite extensively, so it's worth reviewing the content in [Process - Routing MIDI and Audio](#). We're also going to use modulation devices covered in [Tools - Modulators](#).

1. Create a MIDI track but do not add an instrument to that track. This is going to be the “generator” for our track.
2. Within this track, add a single MIDI clip with C3 notes for one bar. Name this the “Generator” track (CTRL+R to rename).
3. Add MIDI tracks and assign the input for these tracks to be the “Generator” track that you have just created above. In the example below I've added a short, plucky sound in the Operator track, a Wavetable pad, and a track with the Electric piano sound.
4. A Shaper modulation device is added to the Wavetable instrument track and the modulation is mapped to the Gain in a Utility device. The “minimum” is set to 50% and the “maximum” to 0% and the rate is set to 1/4 quarter notes so that each beat, the Shaper ducks the gain to produce a pumping “sidechain compressor” like effect.



5. The second Wavetable instrument “Wavetable pre-compressor” is taking the output from the Wavetable instrument but set to “Pre-FX” - before the ducking gain reduction is applied. The audio from this track goes **only** to the Return FX, so essentially we get a nice pad sound, but we never hear the dry sound.

## 7 Recipe - Doing more with less using MIDI tools and plugins



6. In your tracks, you can add MIDI plugins to alter the incoming chord. For example:



You can employ all kinds of MIDI FX here. In this example I have grouped these FX into a MIDI Effect Rack and I'm using Macros to allow me to switch on and change various FX parameters from a single set of 8 knobs.

The “Random” effect is going to add (or subtract) up to 12 semitones shift to the incoming note. Remember that our generator is a single C3 note for one bar. We then quantise this randomness to the C minor pentatonic scale using the “Scale” plugin.

Then we have two distinct “Note Echo” plugins which are going to echo the incoming note at 5 beats and 8 beats delay. Each plugin has the “Thru” setting switched on so we’ll hear the original note and its delay. You can experiment with setting this to “Mute” to mute the incoming note, and just play the delayed version. The device on/off switch on the top left

## *7 Recipe - Doing more with less using MIDI tools and plugins*

of the device is mapped to a macro knob which allows me to switch these devices on and off with a knob or button on my MIDI controller.

The second “Random” plugin adds more randomness to the original note, and all of the delayed / echo output MIDI notes. This allows me to add a little more variation to the output by turning up the “Chance” setting on this device. Note that the “Choices” knob is also mapped to a macro knob, so I can dial up more or less variation.

A third “Note Echo” has a 3 beat delay with a Pitch shift up 7 semitones but with very low Feedback setting so that the pitching up fades quickly. Again, device on/off is mapped to a controller knob or button to allow me to turn this on and off.

Finally a “Scale” device ensures that all this chaos is quantised to a C minor scale.

These are just examples of what you could do, but the concept here is that the generating device can be as simple as you like - a single MIDI note lasting one bar in this case - but these devices then allow you to create a cascade of evolving MIDI note information all within the C minor pentatonic scale. Mapping to MIDI controllers allows me to dial up and back the amount of variation during the track.

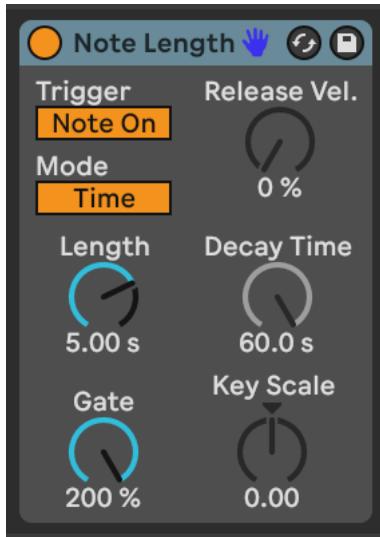
7. In the screen shot above, you may notice that my MIDI Generator track is actually Grouped tracks. Let’s recreate that now. Create a new MIDI track called “Note length” and another called “Arp”. In each of these tracks, set the MIDI input to Track one (the single note C3 track) and set Monitor to “In”.

## 7 Recipe - Doing more with less using MIDI tools and plugins

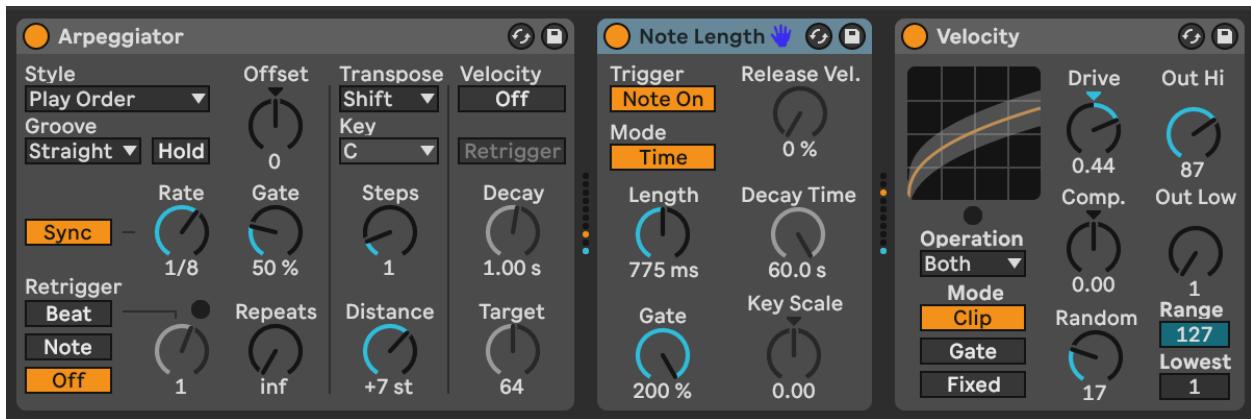


In the “Note Length” track, add a MIDI Note Length MIDI effect. This will be used to take the output from Track 1 “MIDI Note” and stretch out the MIDI notes to last a nominated number of seconds.

## 7 Recipe - Doing more with less using MIDI tools and plugins



In the “Arp” track we’re going to add an Arpeggiator plugin (surprise!) but also another MIDI “Note Length” effect with which we can tweak the note length of the output from the Arpeggiator to suite taste, and a Velocity plugin to add a little variation to the velocity of the output MIDI notes.



Tweaking the Length and Gate settings of “Note Length” will allow us to alter the sound of what is generated from the “Arpeggiator” plugin - note that the Gate setting in the “Arpeggiator” does this as well, but we could potentially map these parameters to a MIDI controller if we wanted to vary the arpeggiator sound coming from this track.

From the MIDI routing screen capture above you can see that we send the MIDI output from the “Note Length” track to the input of the Wavetable pad sound (to give us a nice long chords to play with in Wavetable) while the output of the “Arp” track is being sent to the Electric Piano sound.

In performance of this track, I use a MIDI Controller to change settings of the MIDI Effect Rack in the MIDI note track, which essentially is using me (the human in charge) to dictate HOW MUCH generative randomness happens, but the MIDI effects in the “Note Length” and “Arp” tracks ingest this chaos and make some things happen downstream across a variety of instruments and sounds. I then use the controller to fade up and down the volume of each

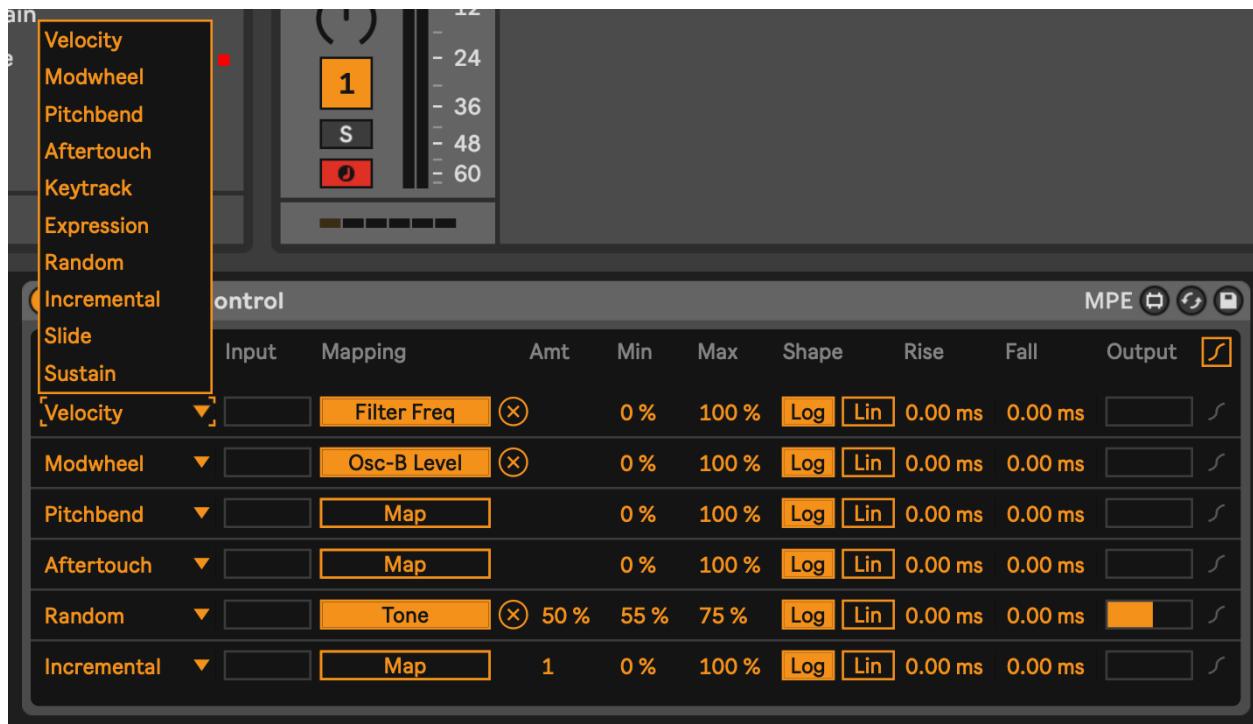
## 7 Recipe - Doing more with less using MIDI tools and plugins

of the instrument tracks to allow me to shape the overall emerging track so that all sounds aren't playing all the time.

8. We have mostly been looking at MIDI **NOTE** plugins here. But there's a much overlooked MIDI plugin called "Expression control" which allows you to map various MIDI incoming signals to **ANY** parameter in Live, including controls of other plugins. In the example below we've mapped various parameters to controls in the Operator instrument just for illustration.



Sure, instruments like Operator and Wavetable have the ability to vary many different parameters using MIDI information, but notice how each of the MIDI parameters on the left hand side has a drop-down menu option. Let's look at what we can do...



There's a wide variety of incoming MIDI inputs which can then be used to map to plugin controls, as well as a "random" input which will select a new value for every MIDI input note, and "incremental" which will increment values for every MIDI input note.

## *7 Recipe - Doing more with less using MIDI tools and plugins*

Also note that the plugin allows you to specify Min and Max levels for each of these - for example the Random input is mapped to Tone in Operator, but the range is limited between 55% and 75% of the values. This kind of constraint can add nice variations in timbre for each note without it sounding too jarring.

Using MIDI “Expression Control” can be useful to modify a variety of parameters in Ableton including some that aren’t easily mapped within an instrument.

9. The basic concept in this recipe - and one that I use frequently - is to take one MIDI part and make the most of it. Using MIDI plugins like this can ensure that the resulting sounds are **related** to the input, without having to be **exactly** the same. Anything you can do to delay, alter, spread out, harmonise and alter the incoming MIDI will add interest to the part. Using these MIDI parts with different instrument parts / sounds and audio effects will add variety and interest.

Because I’m using lots of MIDI “Note Echo” plugins here, the generating input needs to relatively simple - here just a single MIDI note that lasts one bar. If the generating MIDI clip was more busy then I’m sure this would descend into a mess fairly quickly. But the principle extends to other kinds of inputs if you pick and choose the MIDI effects that you apply to it.

The benefit of this approach is that because MIDI effects are altering the inputs, all of the MIDI tracks generated are related to each other and cohesive in their sound, without being THE SAME. This is a key attribute, because we can then use these MIDI tracks across different instruments with different timbres.

10. Press play. Sit back and relax.
- 

You can hear an example of this track here: [https://soundcloud.com/mikeksmith/2\\_single-note-input/s-ijkBeXuelaw](https://soundcloud.com/mikeksmith/2_single-note-input/s-ijkBeXuelaw)

# 8 Tools - Modulators

The MIDI tools discussed in [Tools - MIDI tools](#) are useful to transform, augment and add notes to existing MIDI sequences, but we can also automate changing the value of pretty much *anything* in Live via modulation. [Max for Live provides a variety of modulation devices](#) which will allow you to “map” that automation to device parameters. This will then automate changes on those device parameters without you needing to tweak the values using a mouse or controller knob.

## Key idea

Modulation keeps sounds interesting. In [Process - Balancing the unexpected and the predictable](#) we discuss how repetition balances randomness. This also applies to the sounds you use in your track. Modulation of synth sounds through LFOs and other means is available within Ableton Live instruments, often with some very creative ways of achieving that - such as the Wavetable instrument’s modulation matrix. But sometimes you want to apply modulation to a parameter where an LFO is not immediately available for example to the macro knob of an Instrument or Effect Rack.

Having options where the modulation isn’t obviously cyclical also helps as it adds to the “unpredictable” and helps engage listener’s attention for longer. You could use LFOs with different cycle lengths (in bars or different rates in Hz) or you could use LFOs which are not periodic but which “wander”.

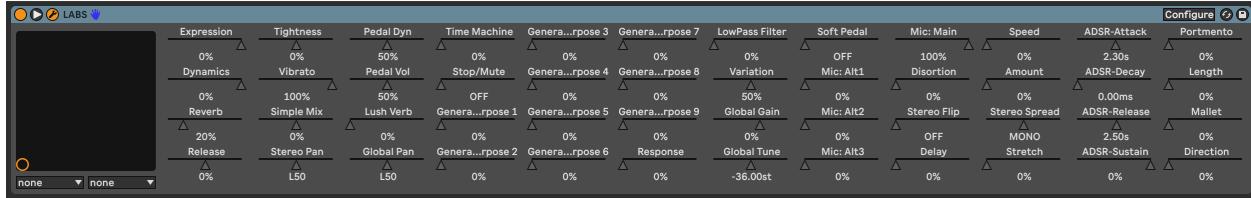
One strength of Ableton Live’s modulation devices (and Max for Live modulation devices) is that you can map modulation to pretty much anything that you can tweak by hand. Yes, even tempo. You can assign the parameter that is being modulated simply by click on the “Map” button in the devices and then click on a knob, switch or fader in Live. Modulation also allows you to have “hands off” performance of generative music by automating volume fades, if required.

## 8.1 Modulators in Ableton Live (general rule of thumb)

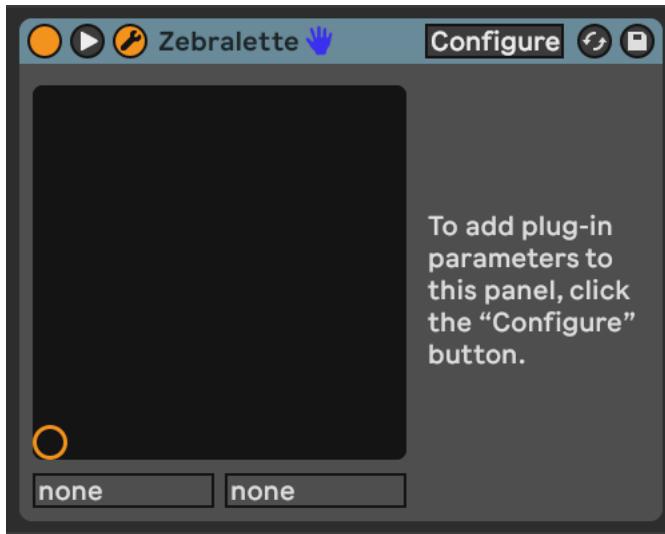
In Ableton Live, most modulators include a “Map” button that you click, then select the item that you want to modulate -e.g in the LFO device <https://www.ableton.com/en/live-manual/12/max-for-live-devices/#lfo>. The beauty of this is that you map the modulation to *the next thing you click*. This applies to any knob, fader, slider, and even third party plugins.

## 8 Tools - Modulators

In the case of third party plugins you can often click on the “Down arrow” beside the plugin on/off button to expand the parameters that can be automated.



If no parameters are presented when you click this button, you may instead see a “Configure” button in the plugin container. When you click on this button and then open the plugin, whatever you click in the UI interface for the device or instrument will show up as a slider in the container interface (see the images below showing the Zebralette instrument as an example).

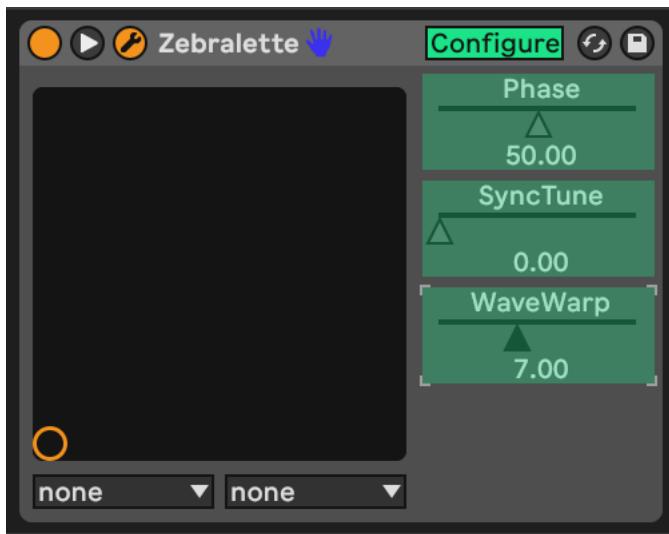


This then opens the UI for the instrument or effect. Now click on various knobs, faders or elements of the instrument or effect UI and Live will capture what has been selected and make these available as items that can be modulated as a list of sliders.

## 8 Tools - Modulators



For example, if we clicked on the “Phase”, “SyncTune” and “WaveWarp” elements in the UI for Zebralette, when we close the UI we’ll see the list of these elements available now for automation or for assignment to Macro knob control.



## 8.2 LFO

Low Frequency Oscillation (LFO) modulation is one of the oldest forms used in synthesisers. In Ableton Live 11 this is presented as a separate **audio** utility plugin ([Max for Live plugin called “LFO”](#)). Because it is an audio plugin it has to come **after** instruments and effects. There is another Max for Live plugin called LFO MIDI if you need it to come before instruments. LFOs are a key element of a modular synth rig. They can be and often are used to modulate pretty much anything. The LFO in Ableton Live can be used in pretty much the same way.

The Ableton LFO is a really neat tool and illustrates some features that we’ll see again and again in other modulators in this section. In the top left of the first LFO here you can see the parameter that is being affected by the LFO. In the screenshot below it’s changing the Filter Cutoff in the Instrument Rack to its left. I have changed the minimum and maximum value from 30% to 80% in order to avoid extreme values of filter cutoff. It’s often a good idea to tailor the range of values that the LFO is working across as often you want to avoid the extremes. In the bottom left of the device you’ll see the Rate at which the LFO is cycling. In this example it’s going to go through a full cycle (of the Sine wave) every 6 bars. If you click on the box with three horizontal lines in the top right of the LFO it will open a page (as seen in the second LFO) where you can assign other parameters to be affected by the same LFO. Note that the second LFO here is changing the Depth of the first LFO and the Offset. This means that the first LFO doesn’t just cycle predictably from minimum to maximum and back, but instead wobbles in a slightly more interesting way. So one LFO might be too predictable, but using two and using one to modulate the other can bring just enough unpredictability to keep the result interesting.



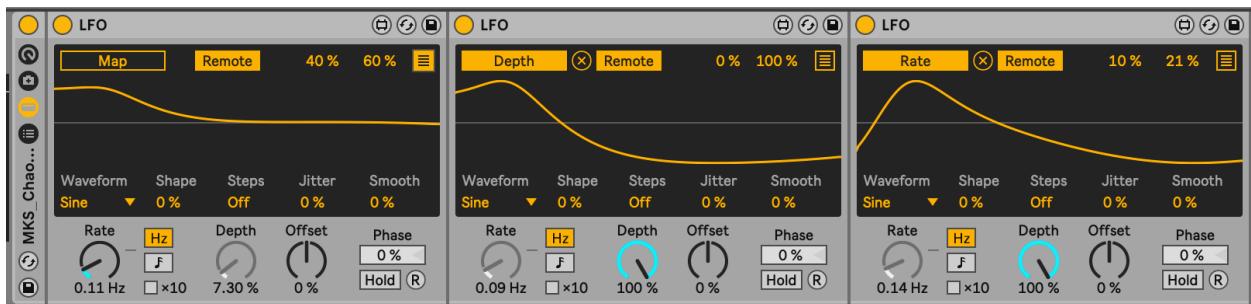
Note that under the visualization of the LFO you’ll see “Jitter” and “Smooth”. “Jitter” adds additional noise randomness to the generated LFO cycle. This can be useful if you want to add a little more random variation over and above the generated LFO values. The “Smooth” option can help smooth out the whole LFO cycle, including the “Jitter”. These “Jitter” and “Smooth” options turn up in various other modulation devices, so it’s good to know what they do.

A basic LFO is a useful modulator, but there are two drawbacks to the standard LFO on its own:

1. Its cyclical nature. Having something modulate within a predictable cycle e.g. 6 bars is fine, but is a little too predictable if the listener is going to be engaged with that sound over a longer period. If that cycle is part of a larger group of modulations moving at different cycle lengths then this might be less obvious.
2. The “Random” setting is not smooth, as it is a “sample and hold” - so the value changes at the nominated cycle time and holds that value until a new value is assigned at the start of the next cycle. This means that the modulated parameter changes in a step fashion rather than smoothly. The LFO has a “Smooth” value (bottom right of the wave display) but this smoothes out with exponential curves, meaning that the value starts changing quickly at the start of the next cycle. This may still be “not quite smooth enough”.

### 8.3 LFO Trio - a chaotic LFO

One trick from modular synthesis is to modulate LFOs with other LFOs, and to modulate those LFOs by other LFOs. (It’s LFOs all the way down). I was looking to create an LFO where there wasn’t a perceptible pattern or period to its output. So I took a second LFO and changed the depth of the first LFO by the second. I took a third LFO and used it to change the Rate of the second LFO. I then took one of the mapping parameters from the first LFO to change the rate of the third LFO (so feeding back from first LFO back to third). What this gives me then is three LFOs which are all behaving in slightly chaotic and unpredictable ways, even using the standard Sine waveform.



### 8.4 Envelope MIDI

The [Envelope MIDI device](#) provides an envelope (shock!) that can then trigger any parameter in Ableton Live. Any time the channel receives a MIDI note, the envelope is triggered. This envelope can operate free of the tempo, with attack, decay, sustain and release in milliseconds, or it can be synced to the tempo. The envelope can also loop if required. The attack, decay and release rates can be linear (when set to 0%) or curved.

This allows modular synthesis like control over all kinds of parameters in Live where an incoming MIDI event can trigger changes in ***anything***.



## 8.5 Envelope Follower

The (Audio Effect) [Envelope Follower](#) device is similar to the MIDI Envelope device, but instead of being triggered by MIDI notes, it is triggered by audio. Adjust the “Gain” of the input until the orange envelope shows a decent range of modulation. By then tweaking the “Rise” and “Fall” parameters you can smooth out the signal if needed. There’s also an option to delay the envelope rather than having it follow the input signal immediately.

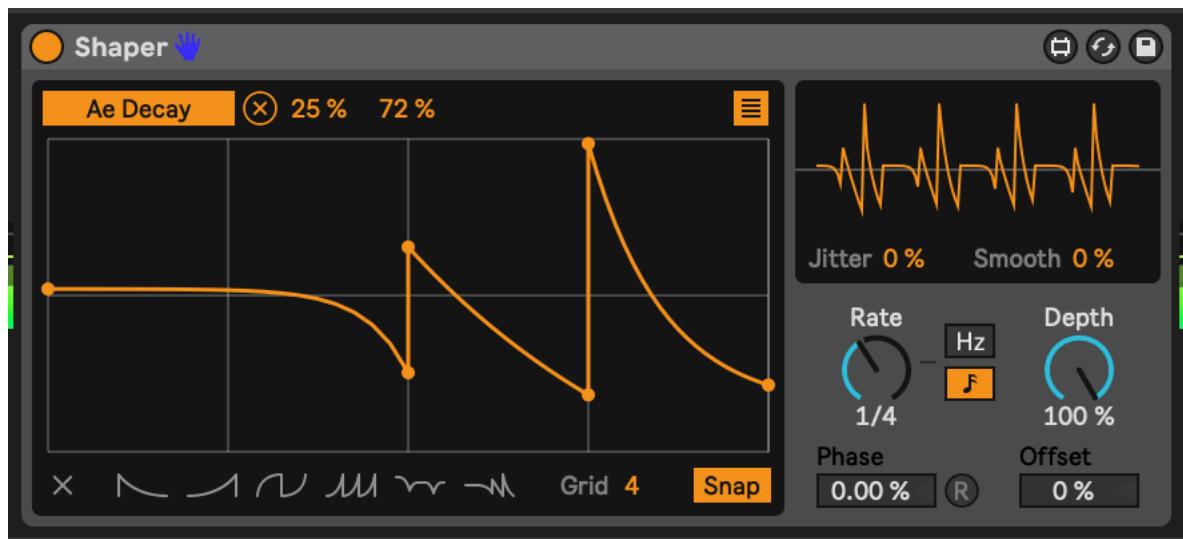
Then, using the usual mapping process you can choose which parameter in Live you want to modulate using this signal. This device allows you to modulate anything using an input audio signal, whether that signal is heard in the Master audio track or not.

The devices above (LFO, MIDI Envelope and Envelope Follower) offer modular synthesis like control over pretty much any parameter in Ableton Live. This is one of the strengths of Ableton Live with Max for Live devices over the standard Ableton version - the level of control and the ability to control devices is really unparalleled. Don’t forget that you can also get modulators to modulate each other... So an envelope can modulate the amplitude or rate of an LFO and an LFO can then modulate the envelope of another input.



## 8.6 Shaper

Like the envelope follower devices above, there is a [MIDI Shaper](#) and an [Audio Shaper](#) device. These allow you to draw modulation envelopes by specifying nodes across a grid then defining how the modulation should change between the nodes. In the screenshot below, the nodes are “Snap”ed to a grid of four equal divisions. The “Rate” setting defines the length of each division compared to the clock, which can be set according to musical divisions, or in Hz time.



## 8.7 MIDI Expression Control

The Wavetable instrument in Ableton Live has a really useful Modulation Matrix where you can map incoming MIDI expression control attributes (Velocity, Modwheel, Pitchbend, etc.) to parameters of the synthesis engine. The [Expression Control](#) device extends that to ANY instrument or effect in Ableton Live. As with LFO modulation mapping, you click on the “Map” button and then click the parameter you wish to modulate. Note that the Expression Control device is a MIDI device so is used **before** instruments while other modulators like LFO are placed **after** the instrument.

Expression Control allows you to see the minimum and maximum percentage of modulation, and the shape of onset / offset of that modulation - either linear or logarithmic (curved). It also allows you to specify the “Rise” and “Fall” rate of change to help smooth out the change and prevent sudden changes in parameter values. There’s also a curve shaper in the top right of the device where you can tailor your own curves for the MIDI input, like compression etc.

The “Random” setting generates a new parameter modulation value **with every incoming MIDI note**. So rather than cycling through a periodic modulation, Expression control can be used to change parameters with each note. Subtle use of this is good for changing the sound of rhythmic parts - selecting slightly different decay and filter cutoff for example.

The “Incremental” setting is also useful in that it increments the parameter value by a nominated amount (1% in the screenshot below) with each incoming MIDI note. When it reaches the maximum, it resets to the minimum value, so is effectively a sawtooth LFO but triggered by incoming MIDI rather than periodic in time.



One thing to note is that Expression Control can assign the same MIDI control message to multiple parameters. You can click on the “down” arrow next to the MIDI control type and select a different input.



As I've said above, the combination of all of these tools are supremely powerful - so you could change envelopes with each incoming MIDI note or use that note to change an LFO setting that modulates something else. Also, be aware that the Expression Control and Envelope Follower can be used *even if* the MIDI or audio from the tracks are not sent to the Master audio output. OR EVEN NOT HEARD AT ALL. So you can use MIDI events to modulate items in sync with the master clock regardless of tempo.

## 8.8 Strange Mod

For a long time, [Dillon Baston's paid Max for Live plugin "Strange Mod"](#) was my "go to" modulation device. It uses chaotic processes - strange attractors - and evaluates the X, Y, and Z three dimensional attributes of a particle within the chaotic system, generating three distinct modulation values (like LFO outputs) which can then be mapped to any parameter as we've seen above with other modulators. The change in the three different modulation values is random / chaotic but because they are mapping the three dimensions of the particle moving in the chaotic system, they do not "jump" to new values. Also the three dimensions are somewhat related - the X value does not jump values, but neither do the Y and Z values. They may change quickly, but if they do this happens smoothly. Note that the "Map" button here allows you to map each modulation value - X, Y and Z - to any parameter in Live, but also each individual value (say X) can also be mapped to 7 other parameters, as with the LFO device. That means that a single Strange Mod device could map out to 24 different parameters. There are 11 different strange attractor algorithms to choose from, and you can randomise the starting point of the process to generate new paths of random modulation. The "R" button ensures that the process is reset each time you start Ableton's transport

## 8 Tools - Modulators

control (press play). This ensures that your random, chaotic process is at least somewhat repeatable! The “Var” setting scales variability in the process - more means that there’s more chaos, while lower values mean less chaos.

It’s this combination of smooth and random that makes this my preferred device for modulation. Slowing down the “Speed” of modulation can be useful to map to sound parameters where you want things to evolve smoothly and gradually.



## 8.9 Dispatch

Dispatch by Cong Burn is a paid [Max for Live modulation matrix](#) which generates 4 different modulation processes (A, B, C and D) which can be modulated by the other processes and then allows you to sum these to another set of modulation values - I, II, III and IV. The Matrix view defines the modulation processes while the Grid view allows you to further combine the A, B, C and D modulation and map these to another 16 parameters.

Dispatch is massively flexible in how you can define and combine modulation sources and types. It can also be quite intimidating at first to know what’s going on. I can recommend reviewing the Cong Burn (Dispatch developer) video on the tool on Youtube: <https://youtu.be/Vlef8VON4Rw>

The vision for Dispatch is that it can be a single modulation device that can send modulation out to parameters throughout an Ableton Live set, rather than having modulators and LFOs scattered throughout the set. If you have the intention of creating generative music where everything is modulated and curated for an extended playback, then Dispatch may be useful to control levels, panning, sends etc. in a single device.

## 8 Tools - Modulators

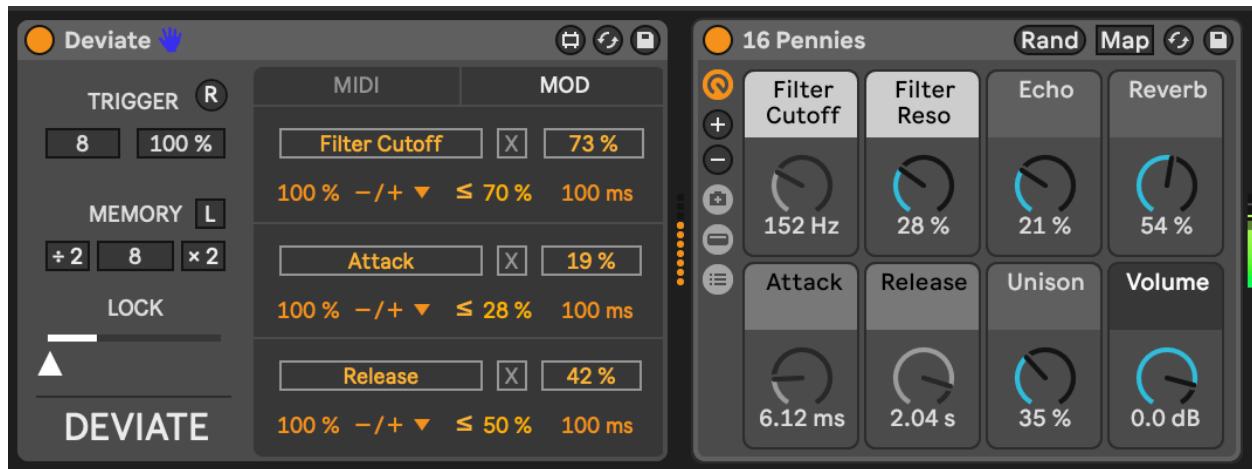


## 8.10 Deviate

We've come across the free [Deviate Max for Live device](#) from [Novel Music](#) in [Tools - MIDI tools](#), but it also offers random modulation control on parameters. You can dial up randomness at various levels in Deviate, and also control the amount of randomness which is very useful in generative music. But it also has a key feature - the ability to monitor parameter deviations in its “MEMORY” and then “Lock” in patterns if you find something you like - this mirrors functionality in Turing Machine modules that exist in the modular synthesis world.

Again, it's worth reviewing [the tutorial video](#) from [Novel Music](#) to learn about all possible features of this device.

## 8 Tools - Modulators



# 9 Process - Tuning Parameters

This is actually a key concept, so ***DON'T SKIP THIS BIT***. It's the difference between a random collection of sounds that wobble about randomly, change notes randomly with very little coherence or musicality, or something that's actually listenable and identifiable as music.

## 💡 Key idea

As with the advice to balance unexpected and predictable, the advice here is to listen to the results of the generative processes and apply your musical ear whether you feel it passes the test of being pleasant to listen to for extended periods without any particular sound, pitch, volume or other modulated parameter catching your ear.

## 9.1 The Problem

We defined “generative” music in [What do we mean “Generative” music?](#) . We defined it as “music (that) should be able to go on indefinitely, with enough going on to make it interesting, but without anything particularly “sticking out” to catch the listener’s ear”. When we use random processes to generate notes, modulate parameters, change the timbre and qualities of sounds, there’s a danger that this randomness will produce sounds that ***do*** “catch the listener’s ear”, or more than that - ***grate*** the listener’s ear.

## 9.2 The Taste Test

How to avoid this? Well, as chefs prepare dishes they continually taste and adjust seasoning so that the end result is neither overly-seasoned nor under-seasoned. Basically they taste and add salt throughout the process. Similarly here, I suggest that you listen at each stage and adjust the amount of randomness, the amount that modulation sources are changing parameters, and whether there’s too much unexpected stuff going on, or whether the track is too static and doesn’t change enough. While listening I find it best to try step away from the Ableton Live session, or minimise the Live screen so that you can concentrate on *listening* and not on what’s happening within the Live set. Some producers listen to their mixes from another room, or with the volume level low, to see if anything “pops out” in the mix. I like listening to my generative music while walking the dog. I make a mental note of anything

that catches my ear - but you can also note things down physically in a book or note for later.

### 9.3 Waving, not droning

In the same way that waves on the ocean sometimes peak together to form very large waves, it's possible for random processes to suddenly produce things that don't sound right. And like the waves on the ocean it's sometimes difficult to tell when this is going to happen. Also like waves on the ocean there may be periods where there is a lot going on and the track sounds "busy" as well as periods where there is less going on and the track sounds quiet and sparse. The trick is to try to adjust modulation settings so there aren't too many "chaotic" periods nor periods where the track is static.

Consider changing the range that the modulation applies - making the modulation more subtle - or changing the period of the random process. If your LFOs are sine or triangle waves that have even number period (maximum effect every 2, 4 or 8 bars) then eventually all of these will reach their peak (or trough) together - after 8 bars - and this will keep happening every 8 bars. Instead if the LFO period is a (large enough) prime number then the peak or trough will take much longer to happen and may never repeat exactly the same. Small amounts of modulations and uneven (and to be honest longer) modulation periods will keep the movement happening in the music without it becoming predictable. You can always use a modulation source to change the period or amount of another modulation source.

### 9.4 Be random, but not *too* random

Constraining randomness can be done up front in modulation - you can change the upper and lower limits of the modulation - and you can apply constraints to MIDI notes via Scale quantisation, you can have sequences that evolve randomly and so over time transform themselves but in a way that isn't jarring. All of these are ways to keep the randomness in check. As with the Taste Test (described above) you should check in with what you're producing at each step to try to keep unexpected and predictable in balance. This should keep your tracks from being too much of one thing or the other.

### 9.5 Tune the parameters

The amalgamation of all of these steps is to "tune" the random and other process to find that sweet spot where things work together to produce something that you and a future listener might enjoy listening too. As mentioned above, try to do this throughout the process rather than just at the end. That way you can balance as you go and achieve something that sounds good in its entirety rather than a combination or collage of different ideas.

# 10 Process - Balancing the unexpected and the predictable

In generative music, we're likely to be using some probability to govern when to trigger certain events, to select clips, to change some aspect of the sounds being generated. The problem with probability is that it generates randomness and in general music composed by humans is not random. Too much randomness can lead to listener fatigue because the "humanity" of the composition or performance is reduced and it's easy to feel detached from the music.

A counterbalance to randomness is repetition. "If you play a wrong note once it's a mistake. Play it twice and it's jazz" as the old saying goes. If your random processes come up with a sequence of notes or rhythms which is played once then it's hard to wrap your ears around it and understand it. But if you play that same sequence more than twice then the listener has a chance to "tune in" to what's going on and contextualise it.

## 💡 Key idea

Balance things that change, which are unexpected to the listener, with things that stay the same and are predictable for the listener.

There are many aspects of musical ideas and sounds that can change - rhythm, timbre, tonality, texture, timing. If all of these are changing all the time it quickly becomes overwhelming. Sudden changes will "stick out" in a way that's typically undesirable for generative music. Evolving changes are preferred as it's possible to "hide" these so that the listener only realises something has changed over an extended period.

Finding the balance between things that change and things that stay the same is important for creating generative music that will bear repeated (or prolonged) listening.

## 10.1 Evolving changes

In Recipe - [Changing chords](#) the clips across and within tracks had different lengths, and the note-wise probability led to different chords with each triggering of the clips. This meant that chords evolved slowly, with changes being gradual rather than all at once. The longer clip length gives the listener time to acclimatise to a particular chord before it changes and gradual, evolving change means that the listener can pick out which part has changed and then settle in to that change, before the next change happens. What we have seen in this

first recipe is how to evolve chords and notes. We could also change other attributes of the sounds or rhythms and in other recipes we'll look into modulation - changing timbre and texture of the sound - and changing rhythms.

## 10.2 Taming randomness - gardening the random processes

In the Western music tradition we have many familiar tropes - chord patterns, melodic harmony, rhythmic patterns - that helps those familiar with those constructs to feel “comfortable” with music that follows those tropes. Almost by definition, probabilistic and randomness in generative music is likely to break some or all of those rules. This quickly becomes “too much” for the listener if **everything** is random.

The key to making music that is generative and pleasant to listen to is to balance unexpected and predictable elements. These could be the notes that are happening, the texture or character of the sound (high-pitched, lower pitched), the volume of tracks, rhythmic patterns. I refer to this process as “gardening” the random processes - taming things that are unruly, messy, too unpredictable or random. You can change the number of repeats of a random process before it changes, or you can rein in the range of a random LFO (or dynamically change the range with another LFO). We can also apply constraints that limit the randomness of note pitches or rhythms, quantising so that it's within the bounds of what we recognise. These constraints - using scale quantisers or rhythmic quantisers - are not necessarily crutches, but rather just guard-rails that help stop what we're producing stray into areas that would “catch the ear” of the listener. Of course, it's completely possible to play with those constraints in order to deliberately play with this listener's expectations.

What's important to me is the balance of the unexpected with the predictable. Carpet weavers will often “disguise” mistakes in a pattern by repeating them and weaving them into the pattern so that they become a feature, rather than a mistake. Similarly, repetition and weaving of “wonky” beats into a pattern of music can bring them back to something more predictable - contextualising the wonkiness.

## 10.3 Deliberate use of repeating patterns - ostinato and predictable rhythmic parts

As well as “gardening” the random processes and quantising the randomness, we can introduce deliberately repeating patterns to help hook the listener into a pattern to contextualise the randomness. Ostinato lines are musical phrases that repeat over and over (and over) throughout a piece of music. Pachelbel's Cannon features a repeating line in the bass instruments (continuo part) which repeats throughout. Over this the upper strings play variations on a theme (pattern with variations). The bass-line in this music never changes - it's there to provide a “ground” theme above which we can weave other parts. Ostinato parts don't

**have** to be the bass part though, they can be higher parts which are predictable and unchanging. Since they are unchanging throughout, it **could** make them boring, but we can play with things like changing the key centre underneath them, or implying new key and chord structures above them (for bass parts). If the ostinato line is a single voice, it's easier to recontextualise these notes than repeating chord patterns, which may get tiring to listen to.

The same concept can be used in rhythmic parts - using a 4/4 kick drum part (notes on 1, 2, 3, 4 of the bar) can provide an anchoring for the listener, allowing you to use much more complex rhythms around this. Even if this kick part isn't very loud or prominent, it can provide enough anchoring for you to layer other "kick" sounds around it to play with rhythms. You can alternatively (also?) use a common snare part with snare hits on 2 and 4 of the bar but have a much more fluid and random kick part. Adding something predictable in one part allows the listener to "hook in" to what's static and provides enough context for what is going on around that.

## 10.4 Repeat yourself

If you look at the Follow Action dialogue box in **Process - Follow Actions** you'll see that it allows you to specify a number of repeats before taking the next action. You can exploit this feature to build in repetition before evolution. Playing the clip 4 times, say, before moving on. This works for shorter clip lengths or for rhythmic patterns, but you may want to avoid this for longer clips.

You could build an ostinato pattern within a clip and variations or evolutions of that pattern and then use the Follow Actions to gradually move through those evolutions, balancing repetition with gradual evolution. This would break the strict "ostinato repeating all the way through" but may allow you to balance predictability with the unexpected.

In future chapters we'll be looking at tools and techniques for changing different aspects of Ableton plugins and instruments. The act of changing something non-destructively i.e. not permanently is called modulation. Modulation tools can be used to gradually change timbres, relative volumes and many other sound attributes.

## 10.5 **YOU** define the parts, the computer picks the order

Again, using Follow Actions, **Process - Follow Actions** you can write a collection of motifs and ideas that you like, or rhythmic parts, or chord changes but then use the randomness of the Follow Action to dictate which part is played next. Combining this with one or more of the ideas above can give you something where the written parts have a familiar structure and are not random with a process that could produce generative music that could last an arbitrary amount of time.

## 10.6 Contextualising randomness using humans

Combinations of the ideas above could give you a framework for generative music piece or performance. If you are skilled in improvisation then it might also become the “second performer” for you to improvise against, with your performance adding the “human touch” and balancing and contextualising what is played by the generative system. So if a motif is played, maybe you echo that motif. Or you could play a chord sequence to drive movement against a random backdrop. Or you could play rhythmic or drum parts again contextualising the random processes.

## 10.7 Play with the “1” - mess with the listener’s perception

In Anna Meredith’s track “Nautilus” the drummer joins the track about halfway through. When he does, it **COMPLETELY** throws me off and makes me rethink what I had perceived for the entire first half of the track, because the “1” downbeat in the bar is suddenly redefined in an unambiguous way and you realise that actually what you **thought** was the “1” was actually an off-beat. It’s very disconcerting. In fact, the beauty of that track is that pretty much the whole way through you’re never quite certain where you are in the bar.

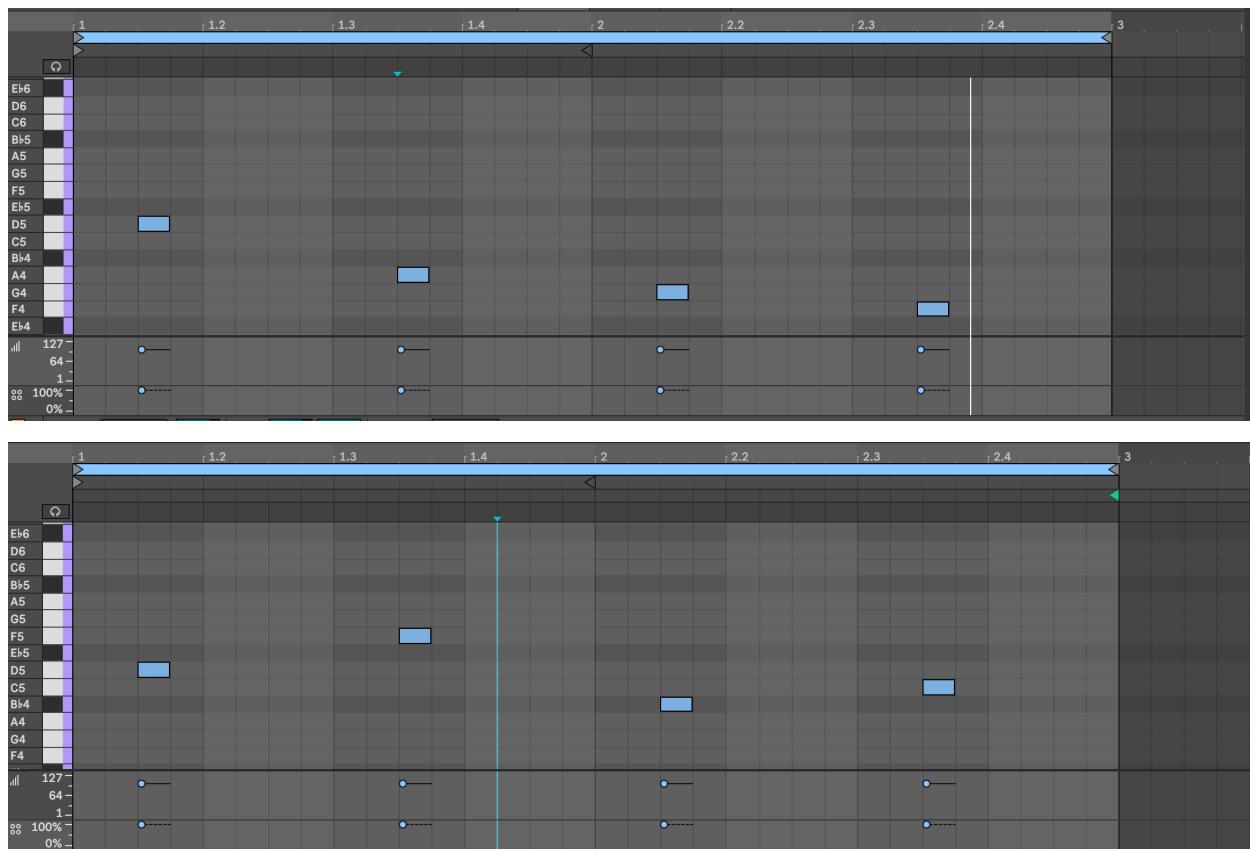
So what can The Lazy Producer do to similarly mess with the listener’s perspective? Well, one way is to use poly-rhythms or poly-meters to slice the bar into different and competing rhythmic parts. If your arpeggiated part is playing “in three” consistently, but the kick on the drum part is playing “in four” consistently, then if that kick doesn’t come in immediately, then the listener will gravitate to where the “1” is for the bassline, but will then have to readjust when the kick enters.

Tonally you might be able to do similar things - your chords and arpeggios may suggest one tonal centre, but that gets redefined when the bass comes in.

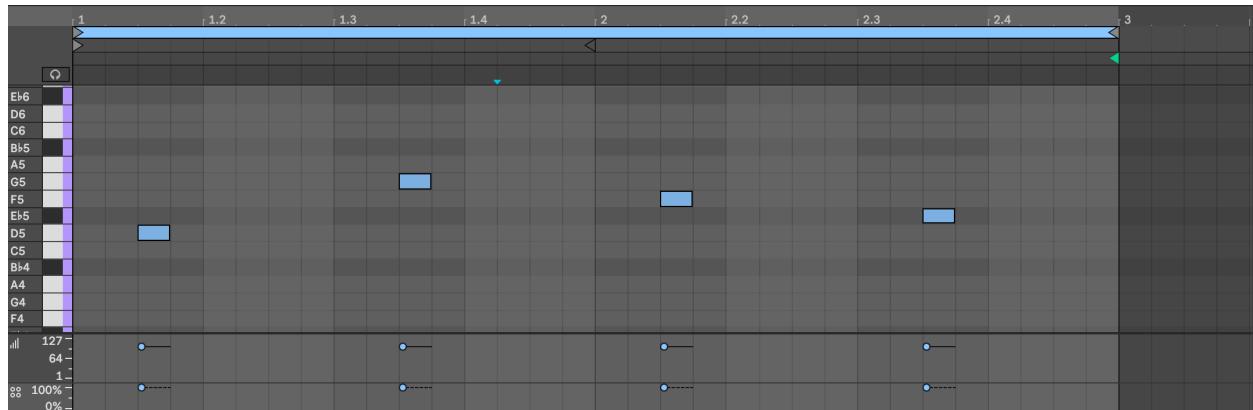
All of these ideas talk about anchoring perception and then playing with that through alternative contexts. You can choose whether that happens subtly or overtly.

# 11 Recipe - Balancing the unexpected and the predictable

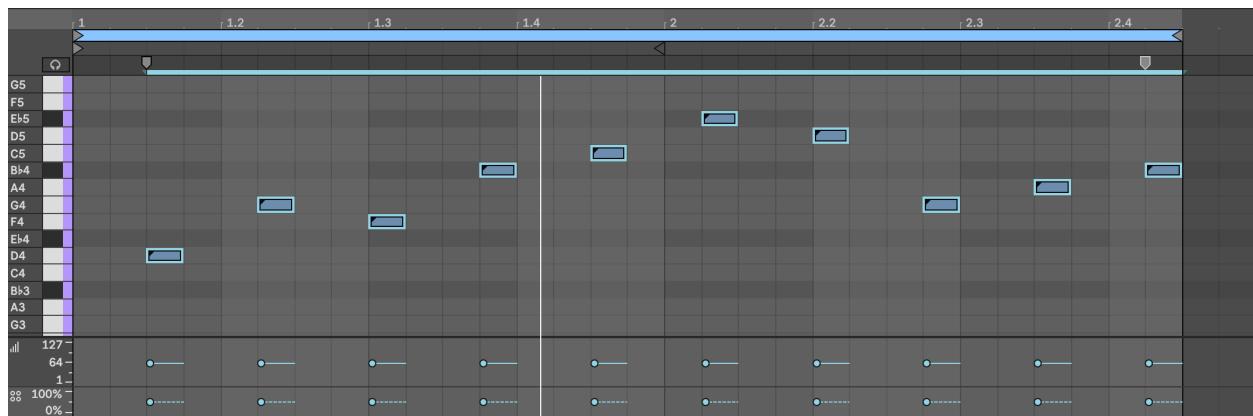
1. Create a short melodic phrase in one MIDI track. I've created three clips, which have Follow Actions set so that the clips advance to the next clip after playing 4x times.



## 11 Recipe - Balancing the unexpected and the predictable



2. On another track, creating a second part that complements the first melodic phrase. Loop this part as well. Ensure that the second part is distinguishable from the first either through having a different tone from the instrument or at a higher pitch. We are going to apply MIDI effects or probability to this second part while keeping the first melodic phrase as our “grounding” part that is predictable.



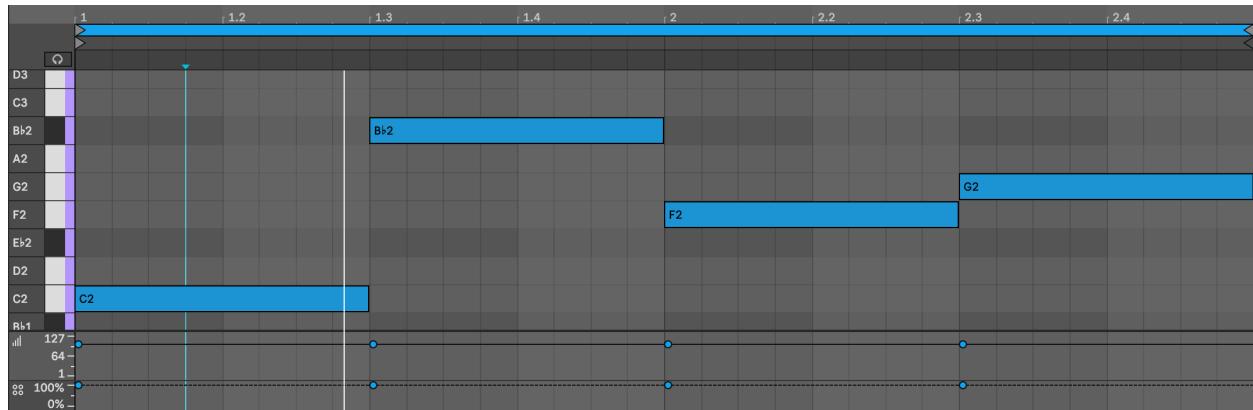
3. You can now choose some options for the second part:

- Lower the probability of the notes in the second part (for example somewhere between 50% and 80%) - so that each time the loop plays you get a different set of notes. With lower probability you can choose to have more notes per clip, as they result will be more varied.
- Add MIDI effects like Random and Scale to this part (see [Tools - MIDI tools](#)) to add some variation on the notes being played. This means that each time the loop repeats you will hear a different set of notes. On its own, this would be “too much randomness”. But because this sits alongside a predictable repeating part, it will sound more grounded.

## 11 Recipe - Balancing the unexpected and the predictable



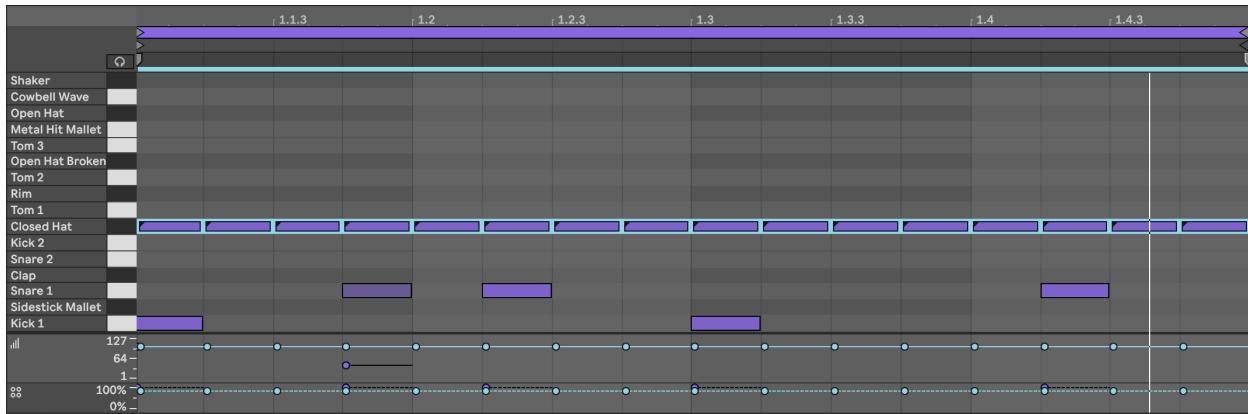
- I also like to add a delay to this second part, either with delay on the track itself or as a Return track. Because the notes are more sparse you can add more effects to them and they will stand out over the repeating, predictable melodic line. If this second part has a higher pitch or brighter tone this will also help. If you are using a Return FX channel with Delay, then you can send the predictable part to this delay at a lower level than the second part. That will mean the unpredictable, higher or brighter tone part “pings out” over the more predictable part.
- Notice that the clip lengths are not 4 bars exactly. This means that when the Follow Actions return to the first clip, the notes will be in a slightly different place in the bar. This will keep the unpredictable part more interesting for the listener.
  - I have also created a little repeating pattern using a pad sound, which loops around. This repeating part adds a “ground” to the track, as it’s completely predictable. This counterbalances the random parts elsewhere. Ideally, to make the track work, this part would ideally be a little longer, but it illustrates the point here.



- Similarly, create a 4/4 kick loop i.e. MIDI notes on 1,2,3 and 4th 1/4 note of each bar. This will be your predictable rhythmic part.

## 11 Recipe - Balancing the unexpected and the predictable

7. Create a Hihat part with notes on every 16th note in the bar.



8. Lower the probability of notes in the Hihat part. You can choose to keep some notes with higher probability to “anchor” things, or make all notes have lower probability (50-80% probability is good). It’s also good to vary the velocity of notes - perhaps notes with higher probability have higher velocity, while notes with lower probability have lower velocity, to provide “ghost notes” on the Hihat part. Feel free to add a Beat Repeat plugin on this part to provide some random “rolls” and glitchy note repeats, if you like that kind of thing. The Hihat part here can be as complex as you like, because your kick part is completely predictable.



9. Press play. Sit back and relax.

---

An example of this recipe can be found here: [https://soundcloud.com/mikeksmith/3\\_balancing\\_predictable\\_unpredictable/s-gPj2EkRvfBh](https://soundcloud.com/mikeksmith/3_balancing_predictable_unpredictable/s-gPj2EkRvfBh)

# 12 Process - Workflow: capturing generative ideas

Setting up an Ableton Live set with lots of MIDI generators, MIDI manipulation and transformation, using that generated MIDI to create a nice collage of sounds that shifts and changes over time is very satisfying. From a performance perspective, once set up, you can press play and let it do its thing. But what if you want to commit the state of the system to share with others. That involves capturing the generated MIDI, modulations and generated audio. This chapter will discuss how you might go about capturing the processes so that you can later release the track.

## 💡 Key idea

I find it useful to separate MIDI tracks that *generate* and *transform* MIDI information from instrument tracks that receive that MIDI information and generate sound. The idea is to capture the generated and transformed MIDI for later processing. It can then be copied into the instrument track and “bounced down” or rendered with the instrument track to capture the sound. I then group all of the tracks (MIDI generator and MIDI devices, instrument track with captured MIDI, bounced down audio) and apply mix FX at the Group level. In this process don’t worry about having more tracks than you think necessary. This will give more flexibility and options in the capturing process. There is quite a large difference between Live sets for performance and Live sets for capturing and committing the performance for release.

## 12.1 In the beginning

When you first start creating Live sets for generative music, the important thing is to have fun with the process. So throw MIDI devices into chains, see what happens, refine and reapply. Route MIDI from one place to another, add more MIDI transformations. Go wild. Play. Inevitably though, at some point, you’re going to want to apply a little order to the chaos. And in fact, even in the creation process taking a little time to plan ahead can open up some new avenues for generating, transforming and routing.

## 12.2 Random processes are random

If you have used any randomness within processes (and if you haven't why haven't you been paying attention up to this point?) then what you'll find is that each time you re-open and play the Live set, you **should** get a slightly different sounding result. It may not be **radically** different, but it is likely to be different enough, that when it comes to mixing before committing to a version of the piece, you find some parts that are thinner than expected, or some where there are peaks of volume or resonance that become problematic for mixing.

In that case, it makes sense to try to capture or "render" the MIDI, instrument and sound design choices so that you can mix and tweak to get towards a consistent output.

In extreme cases you may find, as you capture the output of the random processes that you don't like the output. In that case you might want to refine these random processes - maybe make them slightly **less** random - or tweak settings to try to avoid extreme changes in sound design through reducing modulation levels. But the other option, to be honest is to throw that version away and re-render and re-capture the output to see if the random processes spit out something better next time. Don't feel you have to be precious about the output of some random process. There are many more versions of the random processes readily available.

## 12.3 How to set up Live sets - my experience

### 12.3.1 MIDI Generators

I like to have separate tracks for MIDI generators. These are the "starting point" of a process. They will be generating MIDI notes through random processes, patterns or **Follow Actions**. I like to keep the generators separate from the transformation MIDI tracks, because you may want to send the **same** generated MIDI notes into several transformation chains. This will lead to a more coherent result since different transformation chains ingest a consistent set of generated MIDI input notes.

### 12.3.2 MIDI Transformer tracks

These tracks will take the MIDI generator process outputs and apply MIDI effects and transformations to add additional MIDI notes, information, modulation etc. You could have a virtual instrument / sound generator plugin in this track. If you do that, then you can route the MIDI from this track **AFTER** (MIDI) FX into another track for further transformation, MIDI effects or to another instrument.

### 12.3.3 Capturing the MIDI

To capture the random processes from the generation and transformation steps, you should create new MIDI tracks that “listen” to the output of the generation and transformation tracks. In my Live sets I typically have a lot of tracks where the Monitor is set to “In” or “listening” mode. That passes the incoming audio from another track straight through to the effects or instruments on that track. Having separate “capturing” MIDI tracks allows you to pick up the generated and/or transformed MIDI when you record. This is a good thing to be able to “tidy up” or “garden” the generated MIDI information on the road to release. Don’t feel you **have** to keep the MIDI output from these processes unchanged. It’s just MIDI. Feel free to cut, move, edit, splice until you get something that you prefer.

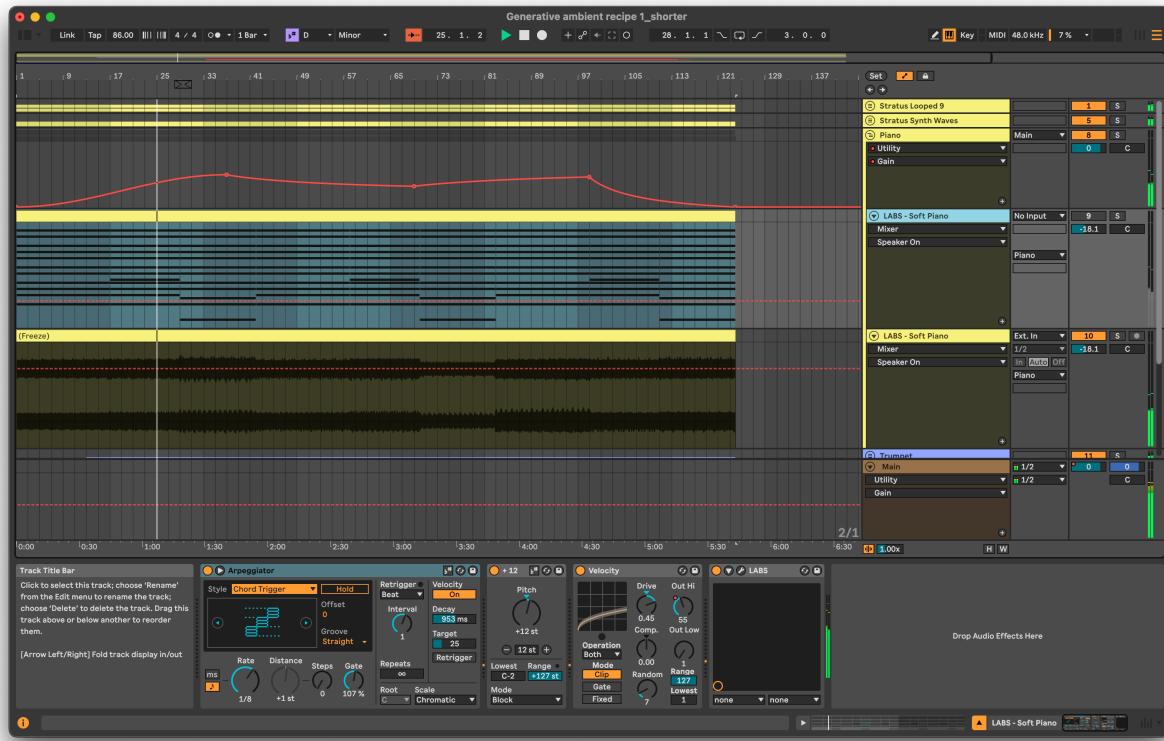
If you record only into tracks where there is MIDI effects and instruments then the **transformed** MIDI is never captured - only the original generated MIDI. This may not be a problem *per se* but when you render out the final audio, you are at the mercy of Live’s internal random processes and you’ll be crossing your fingers that either the result is **the same** as you heard previously (if Live replays the randomness in the same way each time) or potentially different (perhaps if you reopen the set in a later version of Live).

If you capture the generated **and** transformed MIDI in separate tracks, then you can use these tracks as the input for tracks with virtual instruments / plugins in them. Where these tracks might have had MIDI Monitor set to “In” you can now switch that back to “Auto” so that the captured MIDI is used directly as the input for that instrument.

Once you have captured the generated and transformed MIDI it’s safe to switch off the generators and MIDI effects in those tracks. This will ensure that it’s only the captured MIDI that is being used to drive the virtual instruments and plugins.

In the screenshot below I have captured the Follow Action chord sequence, and have added some MIDI effects - Arpeggiator, Pitch and Velocity, before routing these into the LABS virtual instrument. Since the MIDI effects here don’t involve randomness, it’s safe to keep these “inline” with the virtual instrument. If I had used randomness, then it might make more sense to have a separate MIDI track that captures the output of those transformations before sending into the virtual instrument.

## 12 Process - Workflow: capturing generative ideas



### 12.3.4 Rendering / bouncing down to audio

In the screenshot below I've rendered the MIDI part and virtual instrument to an audio file. This then will “bake in” any MIDI transformation - in this case the arpeggiator, pitch shift up an octave, and the velocity curve. Note that I render or “bounce” the output to a new track, leaving the original in place. This allows me to come back later and tweak the original settings or re-render without having to undo many actions.

Since the MIDI input clips with Follow Actions have probability on each note (see [Recipe - Changing chords](#) for more information) then there’s a chance that one or two of the generated MIDI chords will have only two notes (down to the probability choices). By rendering and listening back, I can assess whether I think the “choices” made through probability work with what else is going on or not. Perhaps there’s enough going on when the “thin” chord stack happens that it’s essentially hidden in the mix. If I really don’t like the result, I can re-render the audio.

### 12.3.5 Group tracks to tidy your Live set

With all of these tracks floating about it can be tricky to keep track of what track does what in your Live set. Live’s ability to group tracks is a really useful feature. It’s **your** choice how you group them... You can group MIDI generation and transformation tracks -

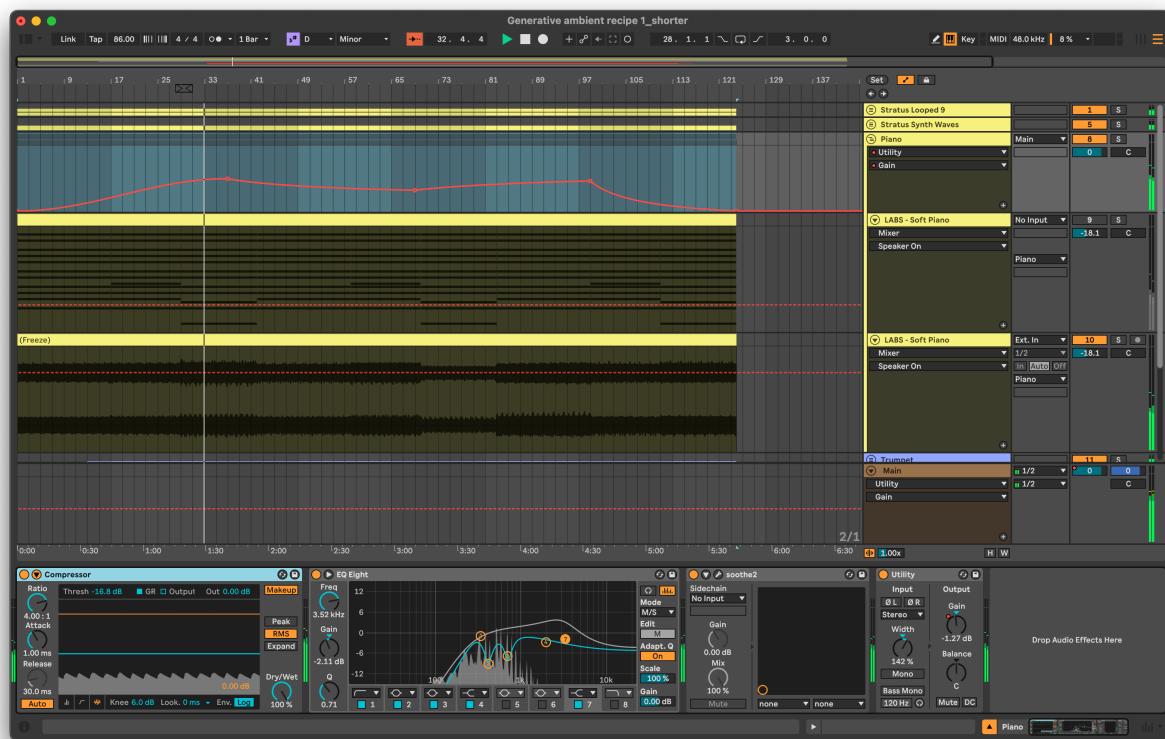
## 12 Process - Workflow: capturing generative ideas

the individual tracks will still send MIDI output to wherever they are routed - as this allows you to “fold away” the generation and transformation processes when it comes to capturing and committing for release (assuming you have captured the MIDI as described above and copied it into the tracks with virtual instruments / plugins).

Another really useful way to use Groups here is to Group the MIDI + virtual instrument and rendered audio together, then apply creative and mixing effects at the Group level rather than on individual tracks.

When I render down the MIDI + virtual instruments, I tend to render without creative and mixing effects as I often need to tweak these later, and it’s often problematic if I’ve committed to certain choices early in the process and baked this into the resulting audio. Instead what I have been doing is to Group the MIDI + virtual instrument track and moving the effects to the Group level (including modulation on Gain - see below) so that the rendered audio is purely MIDI + virtual instrument. This leaves space for me to tweak the creative and mixing effects later when I’m trying to tidy up the overall mix.

In the screenshot below you’ll see that on the Group for “Piano” I’ve got a compressor, EQ, Soothe dynamic EQ and Utility plugins (for automating gain).

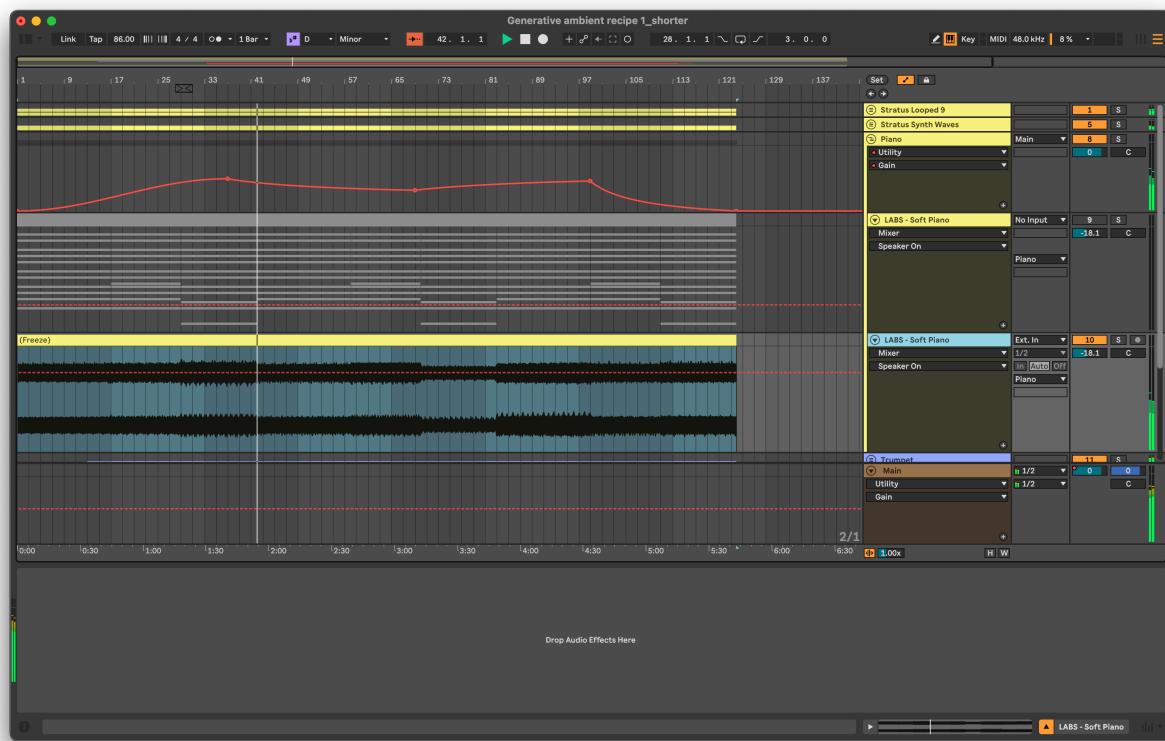


### 12.3.6 Modulation on track volume - use the Utility device to modulate GAIN

If you have gone completely down the rabbit hole of random modulation of your track volumes, it may have been tempting to modulate the Mixer Volume so that tracks fade in and out in a random way. But let me caution *against* doing this. This technique is fine for performance, but when it comes to capturing and committing a generative track, the better solution is to modulate the Gain of a Utility plugin. Modulating gain allows you to separately alter the Volume slider on the mixer to change the *overall* level of a track *relative* to the others. This is a good tip for audio production generally - automate your track Gain, not your Mixer Volume.

### 12.3.7 Turn off what you're not using

Having done these steps - Grouping, moving effects and rendering MIDI + virtual instruments I can now “switch off” the virtual instrument channel and save some CPU (as I’ve done in the screenshot below).



# 13 Tools - MIDI Generators

The MIDI tools discussed in [Tools - MIDI tools](#) are useful to transform, augment and add notes to existing MIDI sequences, but we also need to generate those sequences in the first place. Ableton Live Suite and Max for Live plugins offer a few options to generate sequences of notes.

## Key idea

You can play or program in sequences to be used as the basis for generative music, and we did exactly this for [Recipe - Changing chords](#). But to create melodic or note sequences you may also wish to use a MIDI note generator like those discussed below. All of these devices will create MIDI note sequences - all of them can use random probability to generate note sequences that constantly change or evolve. As we have discussed in the section [Process - Balancing the unexpected and the predictable](#) too much randomness can be difficult for the listener to engage with. Fortunately, most of these plugins allow you to tailor the amount of randomness or to slowly evolve.

In Ableton Live 11 Suite, there are many plugins that will generate note sequences and allow you to specify per note probability. Many of these were developed before Live 11 introduced per note probability in the MIDI piano roll.

## 13.1 Melodic Steps

[Melodic Steps](#) (Max for Live sequencer in the [Creative Extensions pack](#)) is a sequencer that disentangles note pitch, octave, length, velocity and probability. This allows you to sequence these attributes separately. You can define the length of each sequence separately, so you can have all sequences with the same length for predictable MIDI sequences, or change the length of one or more to produce more elaborate patterns. Having sequences that are even numbers of steps will bring you back to repetition faster than having sequence lengths that are based on prime numbers (remember common factors from your elementary arithmetic lessons?). Note that each lane has the option to “roll the dice” and add randomness, either a little (score 2 dice) or a lot (score 5 dice). Also, note that the “Playback” setting can choose to play back at twice the speed or half the speed.

Melodic Steps has been somewhat superseded by the newer [SQ sequencer](#) from the [Live 12 Sequencer Pack](#). See [Tools - Live 12 Sequencers](#).



## 13.2 Bouncy Notes

[Bouncy Notes](#) is a device provided in the “Inspired by Nature” pack developed by Dillon Bastan. This device gives you the ball-drop bouncy repeats triggered by incoming MIDI notes, or with a little tweaking you can get it to automatically produce balls which trigger MIDI notes on a scale, quickly or slowly, resulting in nice echoes and cascades of notes. The balls bounce on the keyboard along the bottom triggering MIDI notes. You can choose whether the balls bounce back up to trigger notes again (“Bounce” settings) and whether the balls bounce off the “walls”. Both of these will keep the balls in motion for longer. Tweaking ball speed, gravity and mass will affect how quickly (or slowly) the balls move around inside the box (“Speed” setting). Higher “Gravity” settings will mean that balls drop towards the MIDI notes more quickly. If you recall physics lessons from school, then you may remember that “Gravity” AND “Mass” affect how quickly notes fall (unless they are in a vacuum). So balls with larger mass fall more quickly under the same gravity. You can affect these via input velocity using “Vel>Grav”, “Vel>Mass” (and in fact “Vel>Hght” and “Vel>Ball”). All together this sets up some physical (pun intended) rules by which the balls move in the space and how they interact with the walls and “floor” which triggers the MIDI notes.

In the screenshot below I have specified that all generated notes should be quantised to D minor pentatonic. I have also chosen for notes to be generated automatically every bar (Drop setting to Auto and “1b”) and last for at most 4 bars (“Lifetime”). At any one time we’ll have at most 6 balls in play (“Voices”). I have also set the note “Quantize” to be eighth notes = 8n. This ensures that the note triggers will not clash with anything else going on in the sequencer that is rhythmic or on the beat. But tuning the “Speed” setting you can tailor the density of notes generated. Adding randomness on direction that the balls are travelling when they are generated “Rnd>Dir” you can generate additional notes in the scale as the balls bounce off the sides and hit the “notes” on the bottom.



## 13.3 Dr Chaos

“Dr Chaos” is part of the [Probability Pack by Sonic Faction](#). This plugin allows you specify a sequence of notes across 8 steps. Notes can be quantised to notes in a scale using the drop down “Scale” setting on the far right at the bottom. The Network panel on the left dictates the order of notes. In this example, Note 1 “C” will be followed by Note 3 “F”. Note 2 “D#” will also be followed by Note 3 “F”. This sequence as defined means that when press “Play” and start the transport in Ableton then the “D#” note won’t necessarily play (since we start at Note 1 “C”). BUT in this case we have a little “Chaos” dialled in to the notes, which adds a little randomness, which might occasionally play the “D#”. Like the “Melodic Steps” sequencer above, Dr Chaos also allows you to define potential Octave, Velocity and Length of each note **and** add randomness to that through the “Chaos” slider. This adds some extra “spice” to the sequence. We can also specify the “Rate” of the sequence, so you can generate faster or slower sequences.

Dr Chaos can store a number of patterns for both the note network (left) and the note attributes (right) so it’s possible to set up sequences that you want to use and play through these, or jump to specific combinations.

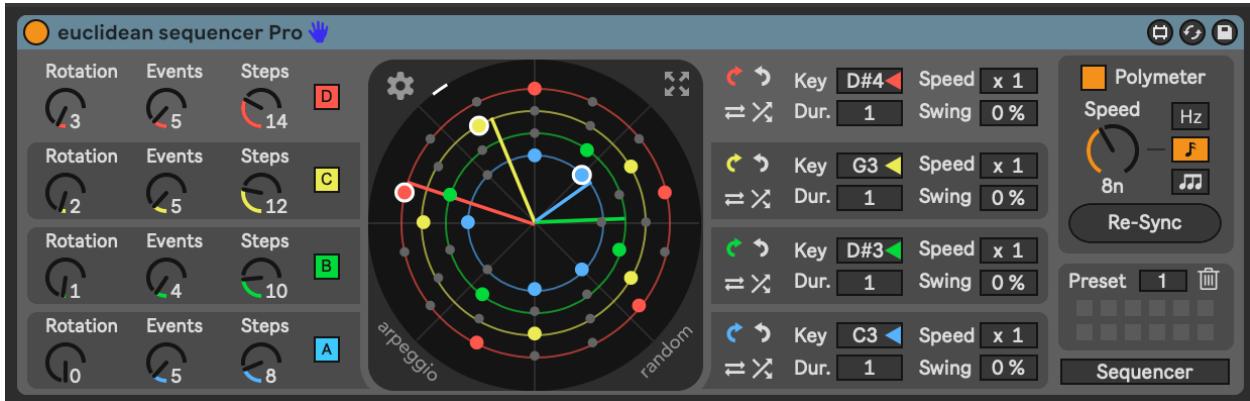
I like Dr Chaos because you can set up networks of notes that will give potentially nice sounding sequences, but randomness can kick in to take you somewhere else instead. It breaks you out of the typical sequencer where the sequence of notes is predictable, even if probability is intervening to stop some notes from happening every time.



## 13.4 Euclidean Sequencer Pro

Euclidean sequencers are commonly found in modular synth systems, and they can produce interesting polyrhythms for melodic instruments and also interesting rhythmic sequences for drum sounds. This [Max for Live Euclidean Sequencer](#) is an implementation of that for Ableton Live. Euclidean sequencers work by defining the number of steps in one rotation of a circle, then allocating a number of events as evenly as possible around the steps in that circle. All sequences start at the twelve o'clock position unless you specify a “Rotation” amount to offset the starting point. You can also specify a “Speed” setting for each of the four sequences, nominate a note that corresponds to each sequence (especially useful when used in conjunction with a drum rack). Particularly useful for generating sequences that are rhythmically interesting.

I have used the [Euclidean Sequencer Pro](#) with non-rhythmic ideas as well. Choose four notes e.g. C3, D4, G3 and A#3/ Bb3. Setting a pattern with more sparse settings for D4 and A#3 can give some interesting patterns and chords. I combined this with the Random and Scales plugins described earlier to add some variation to the generated notes.



## 13.5 Less Concepts

[Less Concepts](#) is a sequencer. How it works is complicated. That it works is down to maths, permutations and combinations of cell values and adjacent cell contents.

*less concepts is rooted in the idea that complexity is just a shit-ton of simplicity, chained together. at its core, less concepts holds 65,536 possible combinations of notes which can be gated, offset, and manipulated to create minimal sequences for improvisation. small changes to a single parameter can bring sweeping or subtle changes. >> seek. think. discover. - Linus Schrab*

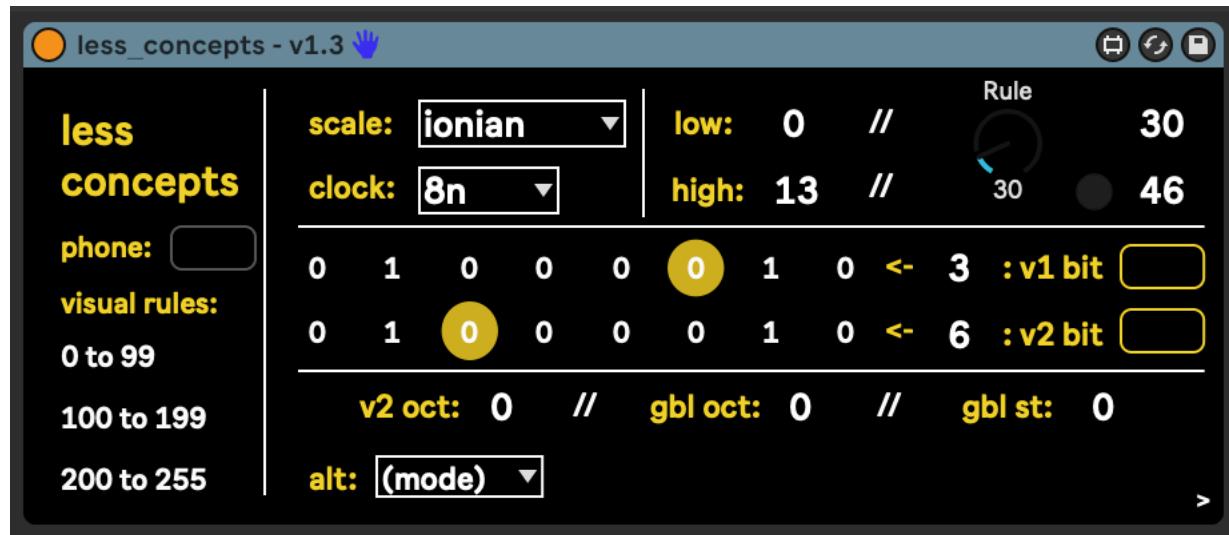
Suffice to say that it generates sequences or bursts of MIDI notes that can be used in generative music. That it's available as a free Max for Live device is down to the generosity and genius of [the musician](#) and [developer Dan Derk](#). There's a PDF manual available

here: <https://lllllll.co/uploads/short-url/q2iJjgbV5JhVNoytmVAuOkVS9f1.pdf> and an accompanying video about the device here: <https://player.vimeo.com/video/408048241>. I think it's best to watch the video about the device, then look at the manual. And I recommend reviewing both, at least a little before you jump in to use the device. Otherwise you might not understand the impact of the different "Rule" values, or what the "Seed" does.

Less Concepts produces MIDI sequences that are less "random" and slightly more melodic. The key items to focus on here are the "Rule" setting, which in the screenshot below is 30, and the "Seed" setting which is 46. Different rules provide different rhythmic / melodic results and the "Seed" setting results in slightly different variations of that rule. For a given Rule and Seed you will always get predictable results, so if you use Less Concepts and fix the starting Rule and Seed you can always guarantee the same result. But one trick I've played with is to use a random LFO to vary the seed every 4 bars (or more) to balance unexpected and predictable, as we've discussed before.

The "low" and "high" settings define the range of notes that will be used in the sequence.

Less Concepts is definitely a sequencer that invites exploration, trial and error. Sometimes the results are surprising, other times they fade away and you're left in silence. If you need further explanation of what's happening under the hood, I suggest you explore the notes that Dan Derks provides.



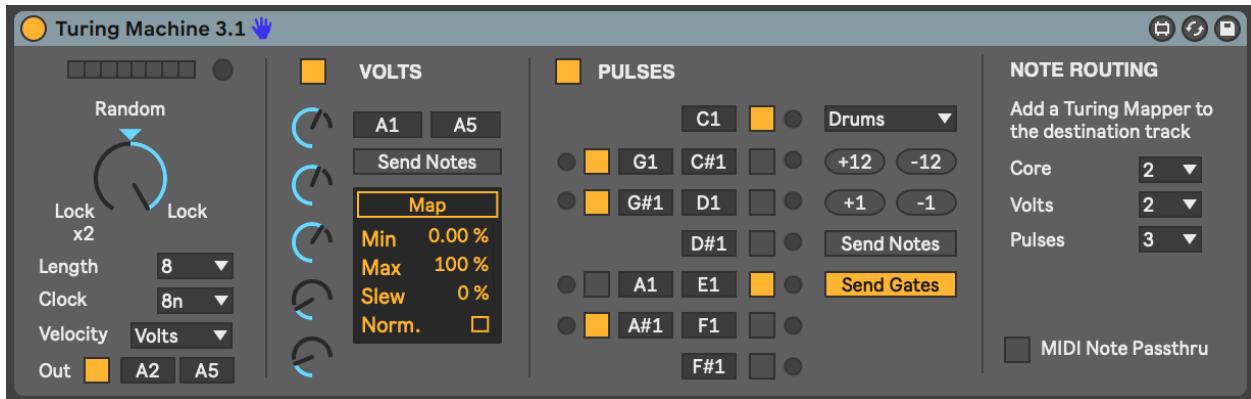
## 13.6 Turing machines

**How** Turing machines work for pattern generation involves things called Shift Registers and moving zeros or ones down a line. Sound & Voltage has an excellent description here: <https://www.youtube.com/watch?v=va2XAdFtmeU> if you want to dig into that and understand better. But it's **what** they do that is of interest to the Lazy Producer. It's a device that spits out random sequences of notes, until you move a dial and "lock in" a pattern. Move the dial a little way back towards the random (middle position) and the device will repeat

the locked in sequence, but add occasional new notes. What's of interest then to the Lazy Producer is the ability to move gradually from random to repeating, predictable sequence and anywhere in between allowing us to balance the unexpected with the predictable.

In the image below the key part is the left hand end of the Max for Live [Turing Machine](#) device. If the dial is set to “Random” then we get a random sequence of notes in the given range (here between A2 and A5) based on the specified clock setting - in this case 8th notes. If we turn the dial completely to the right to “Lock” then we will lock in a repeating sequence of given length - here 8 notes. As we move the dial gradually back to the “Random” setting then we introduce a chance of random changes to the pattern, with the chance of randomness depending how close we are to the “Random” setting. What this means in practice is that we can Lock a sequence, gradually turn the dial back to allow changes to that sequence, then Lock again to “capture” that change. If we turn the dial all the way to the left then we also Lock the sequence, but the repeating sequence will be twice as long i.e. it will take twice as long to come back to the start of the sequence. Because what is generated is a random sequence, we might want to apply a Scale plugin to this to quantise the note outputs and conform them to a particular scale. We may also wish to use [MightM4L \(MML\)](#) [Probability Gate](#) device to “thin out” the stream of notes from the Turing Machine.

The Max for Live device shown below includes additional features to map output “voltage” signals (modulation) from the Turing Machine to devices in Live, and to use the Gate information from the Turing machine to trigger devices like drum machines. These extensions to functionality are still driven by the same essential random processes and “locking in” sequences discussed above.



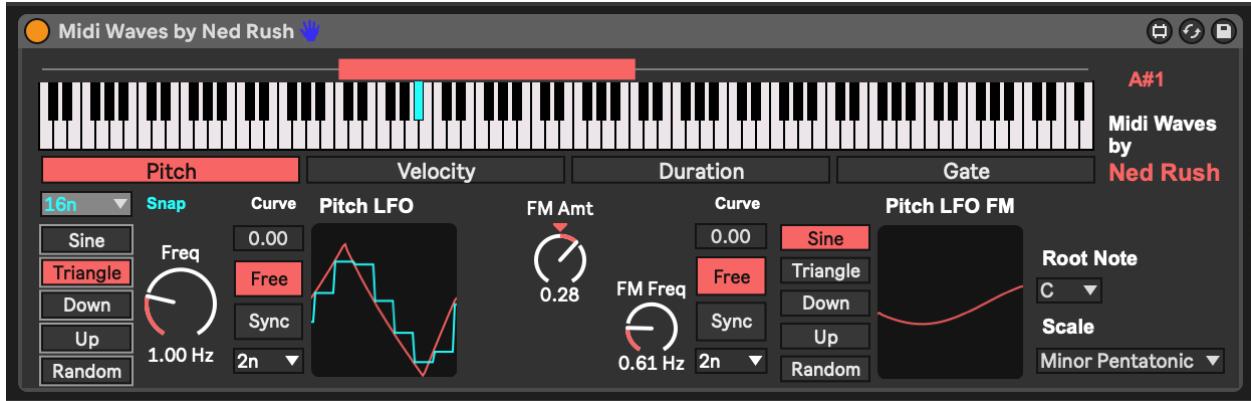
For Live 12 Philip Meyer has developed a Max for Live [Turing Machine](#) Generator device (<https://www.youtube.com/watch?v=fwBIVaAaWog>) that you can use to create patterns of notes. More on this in [Tools - Live 12 Sequencers](#).

## 13.7 MIDI Waves

[MIDI Waves](#) is a paid Max for Live plugin from Isotonik and Ned Rush - <https://www.youtube.com/watch?v=NTDBqpCu21M>. It is a very useful MIDI sequencer based on the

principle of quantising base LFO waveforms through sample and hold. The base LFO waveform (in the screenshot below it's a Triangle wave) runs at a specified rate, either Hz or Synced to Ableton's clock. Pitch values are sampled at a separate rate (here 16th notes) to generate ascending and descending note pitches. The rate of sample and the rate of the base LFO dictate how closely the generated pitches follow the base LFO but can sometimes result in sequences that follow arpeggio like patterns, but not quite (due to slight differences between the base LFO and the sample rate). Note that a second LFO is available to add into the base LFO as a frequency modulation (FM) and you can choose the frequency, shape and amount to be added to the base LFO. This can add more random variability in the pitch sequence generated. Note that the FM LFO is added **before** the blue “sampling” process. So the Sine LFO here is adding a little positive and negative to the base triangle LFO before sampling, resulting in pitch sequences that change subtly each time. Similar processes can be used to generate Velocity, Duration and Gate information. Gate in particular allows you to define the rate at which notes are generated, independent of their pitch. What can result is slight changes in the rhythms of the generated notes, right up to their being clusters of notes, rather than a steady stream.

This technique is sometimes employed in modular synths if a more traditional sequencer is not being used, as all you need to drive it is a base LFO and a “sample and hold” random generator. In modular synths too, pitch sequencing and gate sequencing are often separate processes and it’s nice to have that option here too in the MIDI Waves device.



## 13.8 Midivolve

The [Ableton Max for Live pack and device with Coldcut - Midivolve Pack](#) is a paid plugin for Ableton Suite. It works by randomly adding variation to a MIDI sequence within the device OR it can import a MIDI clip and then create variations using that clip as the original material. That variation can act on MIDI notes, velocity, duration, density (how many notes are playing) as well as being able to map two different parameters in Live. The “A” column defines which parameters experience variation on the next “Evolve!” trigger (in the screenshot below this is set to automatically evolve all parameters once the clip has played through 3 times). If the “I” column is checked then the last evolution acts as the basis for

the next variation change. Note that variation amount can be specified via “Ammt” and this is separate to “Chance” of the variation happening. Chance is the probability of the value being varied / evolved **for each note / step**. This opens up a large number of possibilities to vary input sequences but to balance the unexpected evolution with a repeat of the evolved pattern before the next change is made i.e. let the listener get used to each chance - being predictable, if only for a finite period. Variation can be constrained through the Scale settings.



The Mididvolve pack also comes with some very interesting presets curated by Coldcut that provide some interesting examples of what can be done with the tool.

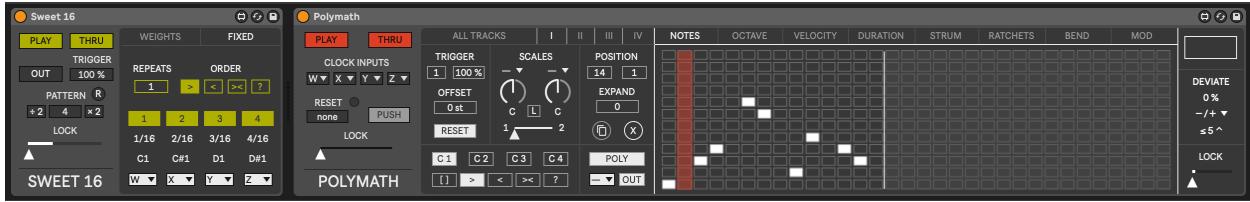
### 13.9 SEEDS - Polymath and friends

[Seeds from Novel Music](#) is actually a collection of sequencing plugins that offer modular synthesiser style control over clocks, note sequences, scales, and much more. This is a paid pack of Max for Live devices, but if you are interested in a collection of tools for modular style sequencing with Ableton Live there is a HUGE amount of potential in this pack. In fact, I can't do justice to the full set of features available, and I can only recommend that interested users look into the [tutorial videos prepared by Novel Music to describe their functionality](#).

In the screenshot below I'm using the “Sweet 16” clock module from SEEDS which provides four different clocks which can be used by Polymath. Each clock proceeds at a different pace, all driven by Ableton Live's master clock i.e. Ableton provides the “1/16th” pulse that drives the clocks. Each clock is allocated a label - here “W”, “X”, “Y”, and “Z”. This helps to identify the clock drivers in other devices. The clocks can be used individually, but they can also be combined to move the sequencer forward in interesting ways.

Polymath is a four track sequencer. Each track specifies notes, octaves, velocity, duration, strum, ratchet, bend (pitch bend) and mod. The sequence length is set for the track. If you look to the right the “Deviate” setting allows you to add some random probability to each element of the sequence as we discussed in [Tools - Modulators](#). This allows you to specify elements of randomness to be included, but also a “Lock” slider that allows you to lock in to a sequence that you like and “turn off” further introduction of randomness.

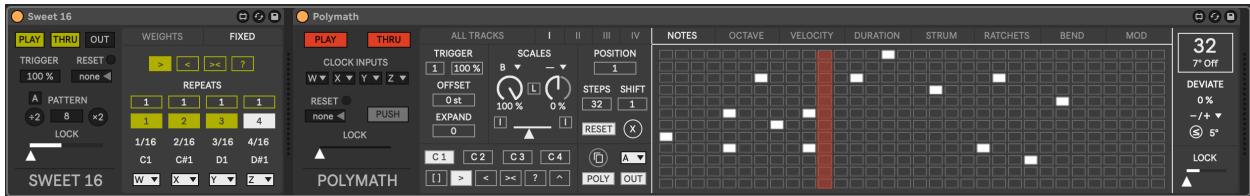
## 13 Tools - MIDI Generators



Polymath has a few neat tricks up its sleeve. It allows you to set scales for each track (and in fact across ALL tracks) and you can set two different scales then use a slider to begin to introduce notes from Scale 2 in the sequence even though the majority are from Scale 1. When the slider is all the way to the right, then Scale 2 is 100% in effect.

One of the great strengths of Polymath and the SEEDS pack is that many of the elements are defined then used across the devices. So the Clocks we've discussed above can be used across devices individually or combined to give interesting rhythmic patterns.

SEEDS also includes a very useful concept and plugin that can send output to and from various devices and clocks. The “Hub” plugin receives and sends MIDI information from one track to another. So it's possible to have generative plugins on one track which send MIDI information to a completely different track, which can then be processed independently. Similarly you can assign the output of a sequence to a “HUB” (e.g. Hub A) and then use the notes of that sequence to offset the notes in another sequence, as is shown in the image below.



This technique allows you to specify two sequences (ideally one shorter than the other) and by offsetting the notes in one sequence by the notes in another sequence you get an ever evolving sequence, with some notes repeated, but some changing.

The “Expand” setting adds additional notes to the sequence, always honouring the Scale set for the sequence. This can be dialed up and back so it's possible to add additional notes and then go back to the original sequence.

Each track can run to a different clock, so it's possible to set up four different sequences that proceed at different rates. Very handy for generative music. Also, the sequences can run forward, back, forward-then-back, and randomly. So somewhere in there you should be able to find a sweet spot to generate some new ideas.

### 13.10 Others I haven't mentioned

Tom Glendinning / ELPHNT is keeping a curated list of Max for Live plugins he uses in his music which includes some excellent sequencers. If I haven't touched on your favourite Max

## *13 Tools - MIDI Generators*

for Live sequencer here, you should [check out Tom's list](#) and review what he's presenting.

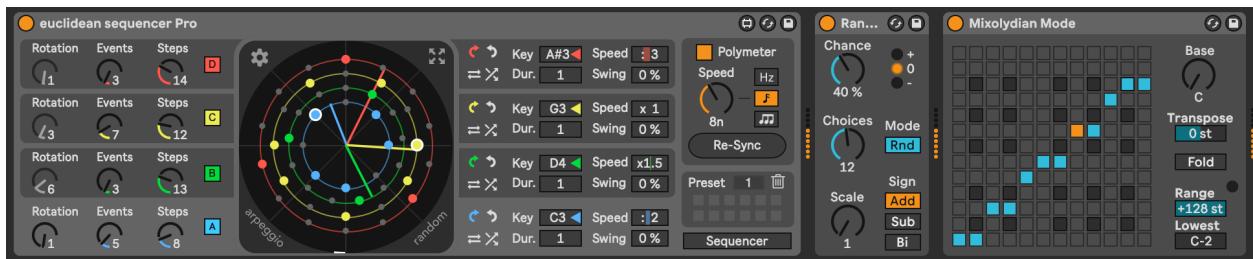
For example the MDD\_\_\_\_\_SNAKE device is based on the Make Noise René modular sequencer. Tom has provided a [very helpful introduction here](#).

# 14 Recipe - Using MIDI generators

In this recipe we're going to use a MIDI generator tool - Euclidean Sequencer Pro - to generate the initial MIDI information which is then fed into additional tracks using techniques discussed in [Tools - MIDI tools](#) and [Process - Routing MIDI and Audio](#).

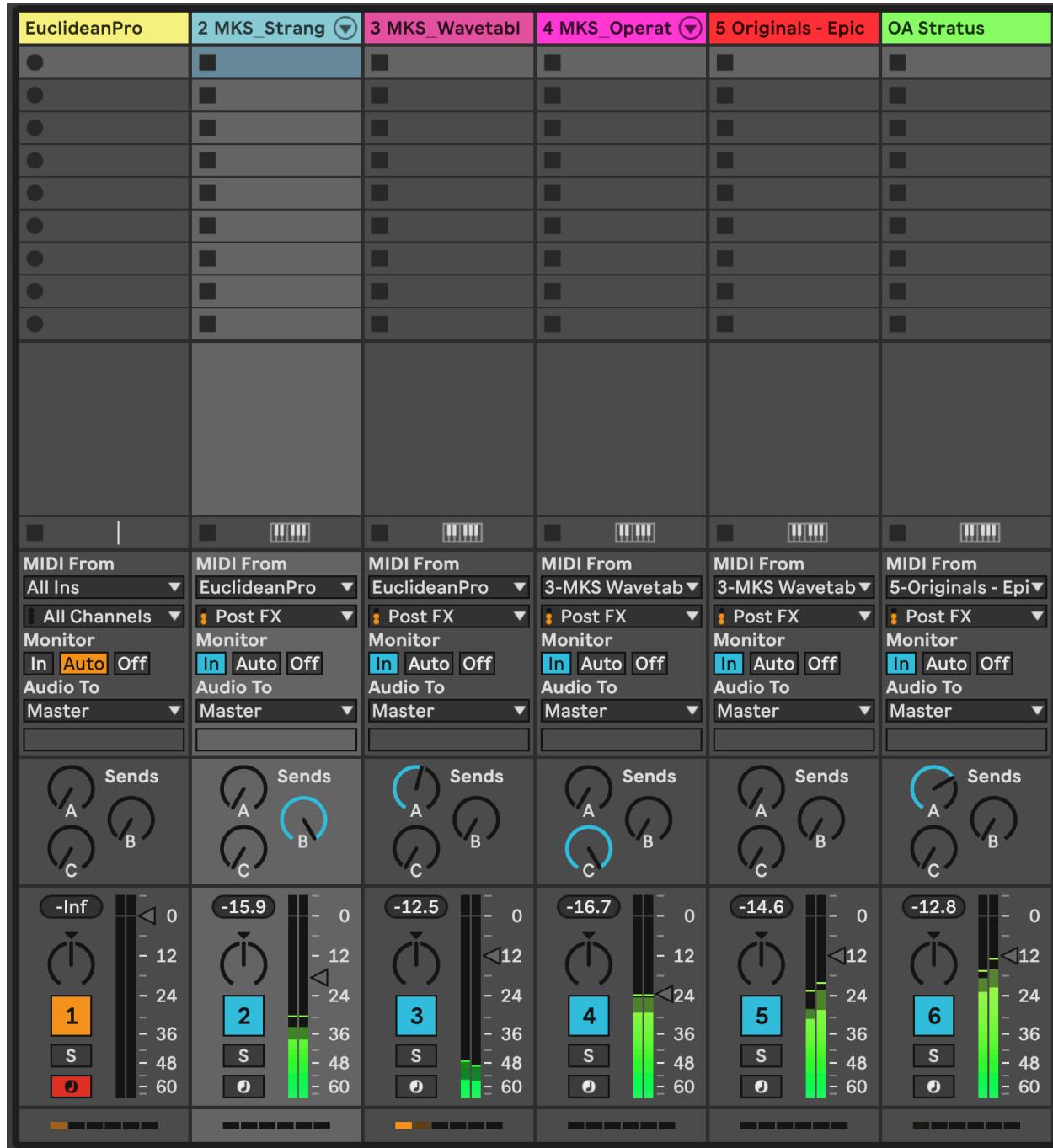
1. Place the Euclidean Sequencer Pro generator in a track. Euclidean Sequencer Pro is a Max for Live device which can be used to generate polyrhythmic sequences of notes, and is typically used for rhythmic patterns and because of this, by default it only generates 4 note values. Because we're going to use it here for more melodic output, we're going to add Random and Scale MIDI devices so that we generate a wider range of notes.

Feel free to set up the Euclidean Sequencer Pro device to generate a variety of pulse Events across a variety of Step values for each note. It's nice to mix up odd and even Events, Steps and Speeds across the 4 generators. This ensures that we don't just get a repeating pattern, although it might also be interesting to explore a combination of predictable and more unexpected patterns (balancing the unexpected and predictable!). We can also use modulators to shift the Rotation / offset of the sequences to add additional unexpectedness if that's helpful.



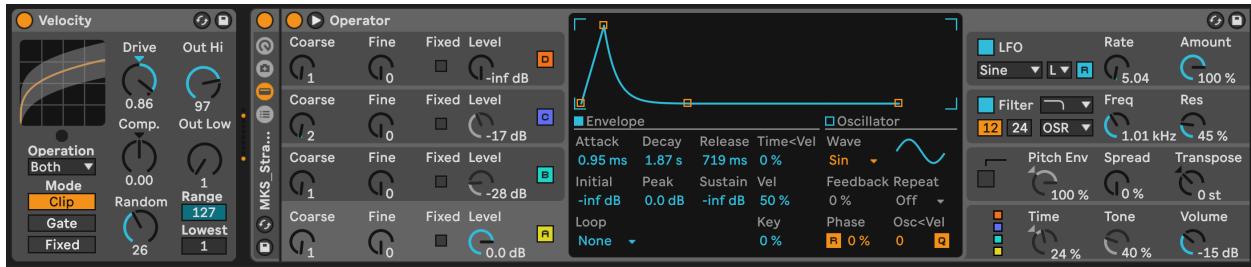
2. We're now going to use many of the techniques previously seen in [Tools - MIDI tools](#), where we take the output from the Euclidean sequencer and then route it to a number of different synth engines while also applying some additional MIDI tool processing.

## 14 Recipe - Using MIDI generators



First, let's take the “raw” output from the Euclidean Sequencer and route it into a MIDI instrument track using a plucky Operator sound - could be a simple sine wave with no sustain and fairly quick decay.

## 14 Recipe - Using MIDI generators



3. Create another MIDI instrument track where we're going to take the same Euclidean Sequencer Pro MIDI information, but now use the MIDI Note Echo device to delay the input by an amount (I'm using 16 16ths = 1 bar). The key thing is to MUTE the original MIDI information. Also add a Note Length MIDI device to turn the incoming MIDI notes into chords. We can then use these delayed and "smeared" MIDI chords with a pad sound. I'm using Wavetable, but you can choose a pad sound from any virtual instrument.



4. Create another MIDI instrument track and use the MIDI from the chord "pad" track we've just created as the input. Be sure to set the input to "Post FX" so that it will apply the MIDI device effects to transform the MIDI **before** routing it into the new track. In this track, we'll add an Arpeggiator MIDI device. We can use a bell like tone for this track. I'm going to use Operator again, with some modulation applied to the second modulating oscillator (which will then vary the tone of the main Sine wave carrier oscillator). I'm also going to apply modulation to the rate of the Arpeggiator device. You can try out various settings here, including using an unsynced or "Free" running Arpeggiator rate. You may need to tailor the amount or range of modulation to get Arpeggiator rates that don't swing too far between VERY quick and VERY slow.



5. Feel free to continue adding additional tracks to bring texture and a variety of tones to your track. I've included a "choir" and textured pad tracks. Experiment with where these take the incoming MIDI information from, and how you can manipulate the MIDI further using a variety of MIDI transformation devices.

## *14 Recipe - Using MIDI generators*

6. You could also experiment with other MIDI generators discussed in [Tools - MIDI Generators](#), such as Bouncy Notes, Turing Machine, Less Concepts, etc. The MIDI generating tools provide the raw stream of MIDI notes, but the effects here are used to provide the texture around that sequence.
  7. Press play. Sit back and relax.
- 

You can hear my version of this track here: [https://soundcloud.com/mikeksmith/4\\_eu-clidean\\_midigenerator/s-qYZKj7UnXXG](https://soundcloud.com/mikeksmith/4_eu-clidean_midigenerator/s-qYZKj7UnXXG)

# 15 Tools - Audio Generators

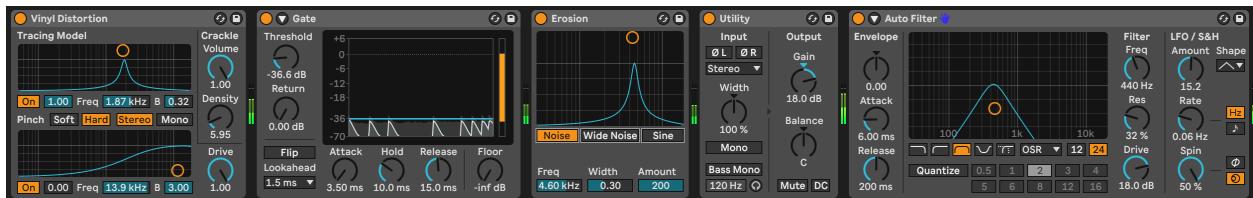
Create something from nothing with these audio generators. These aren't so much "instruments" - they do not need MIDI input to drive them, and will make noise as soon as you press "Play" on the Ableton transport. That's both a good thing and a bad thing - you may want to start a set with the faders down! These devices can be used to power a drone set very effectively though - just route them through additional effects and/or resonators to create some rich tones.

## 💡 Key idea

MIDI generators (as discussed in [Tools - MIDI Generators](#) produce MIDI notes which can be further processed via the MIDI plugins that we explored in [Tools - MIDI tools](#) and then played through instruments. But Ableton Live also has some sound generators that will produce sounds without the need of MIDI input or in fact *any* input. We just need to think a little creatively and use a generous number of Audio plugins to process the sound into the shape we would like. These sound generators are more likely to produce drones and textures than melodies, so may augment some of the tools previously discussed.

## 15.1 Pinging resonators / resonant filters

There's a technique in synthesis where you "ping" a resonant filter with a click or burst of noise which provides the "exciter" to generate a tone. This is relatively simple to achieve in Ableton Live using stock effects in Live Suite. We use a burst of noise from the "Vinyl Distortion" plugin which emulates the crackles and noise from a vinyl record, gate this to only allow the crackles to come through. The "Erosion" plugin accentuates the noise around this crackle and the "Auto Filter" specifies a frequency band that will pass through to the later resonators and effects.



Hainbach has a great video explaining the technique here: <https://www.youtube.com/watch?v=qJRXxkswXdg>. Ned Rush also routinely uses the Vinyl Distortion to ping other devices e.g. <https://www.youtube.com/watch?v=sIWk44aAYeg>.

In our case, instead of using feedback with filters, we can exploit Ableton Live's own resonator plugins: "Resonator", "Spectral Resonator" and "Corpus". In the image below we're using the "Spectral Resonator" to generate some pitched tones. We're feeding it notes from an external MIDI clip which uses the same note probability techniques explained previously. This then feeds into the "Resonators" plugin which provides additional pitched resonance, and the "Spectral Time" plugin which smears out this pitched resonance so that it's not a short burst of pitched information.

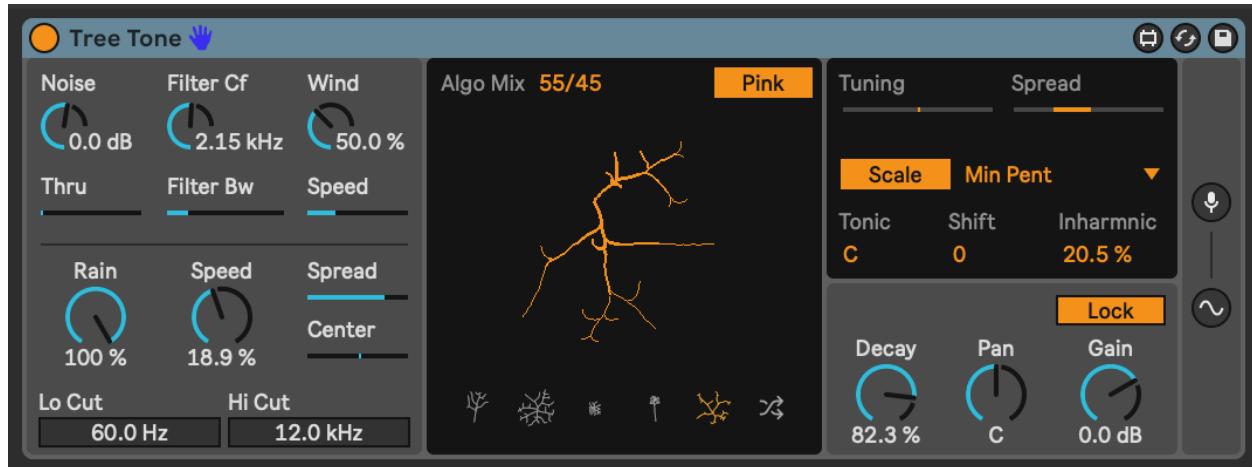


### 15.2 Tree Tone

From the [Inspired by Nature](#) pack by Dillon Bastan the **Tree Tone** generator is primarily a resonant drone generator but with some additional bells and whistles. Literally. The "Tree" in the central panel has "branches" that define resonators / resonant frequencies. Thicker, longer branches have lower pitch and louder volume. Thinner branches have higher pitch but lower volume. In the image below the "Tree" has fewer large branches and so would be expected to have some lower drone sounds. The position of the branches to left or right is their position in panning. The "Noise" amount and filtering determines how much these resonators are excited by the noise. The "Wind" provides additional movement and modulation to the system - affecting volume and filtering. The "Rain" setting "pings" the resonators as with a "raindrop" (or mallet). More "Rain" and "Speed" will ping the resonators providing chimes. The "Algo/Mix" setting will affect the overall pitch of the system - lower values being warmer or darker, higher values brighter. Tuning the system will ensure that frequencies and "pings" from raindrops conform to a scale. "Inharmonic" will add some inharmonic frequencies or more "clangerous" sounds.

This is a deceptively useful generator, as it can be used to create beautiful drones, or you can switch off "Noise" and have just the "raindrops". Combining this with some additional resonators or effects can create an evolving, generative ambience just on its own.

Two Easter Eggs to Tree Tone exist: The top "microphone" button on the far right allows you to route ANY audio signal through Tree Tone and excite the resonators. So Tree Tone can be used as either as a generator OR an effect (similar to the Corpus device). Below this is the "Sine" button which opens modulation dialogue where you can device two LFOs and two envelopes and then route these to parameters of the device. So you don't need to stick with a static sound, and the envelopes can be triggered by incoming MIDI information.



## 15.3 Harmonic Drone Generator

Does exactly what it says on the tin. This is a drone generator from the [Drone Lab pack](#) in Ableton Suite. In the Harmonic Drone Generator you can set the root tone, and then the resonant frequencies of other resonators. When the button is lit orange above the resonator it will be heard. Below the resonator you can choose to add or decrease the resonant pitch (up to 10Hz either above or below). Note that resonant pitch tuning can be set to “Just” intonation or “Equal temperament”. “Just” pitch divisions result in more harmonious noises, but don’t necessarily correspond to tone pitches. So while “Just” will give you better sounding drones, don’t be surprised if it sounds “off” compared to pitched notes from other sources...

Again, this is a nice generator to use as the fundamental source for a textured drone session. You can feed the output of this device through other effects to manipulate, distort, and augment sounds from it. You can also “play” the device using MIDI input. The “Glide” option gives a nice glide effect as the resonators re-pitch to the new incoming note.



## 15.4 Cellular Degradation

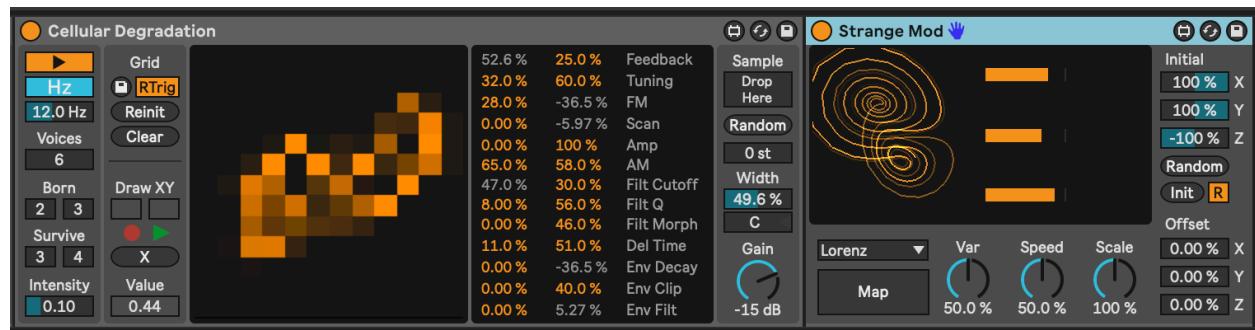
This is a [paid plugin by Dillon Bastan](#) which features six tone generating synthesis engines with a generator mechanism based on cellular growth and decay. If you're familiar with Conway's "Game of Life" then this concept might be familiar, but basically cells are generated and rules govern whether that cell will continue to survive or decay depending on the state of neighbouring cells. So if a new cell has surrounding cells which are alive, it will "survive", but if there aren't sufficient live cells neighbouring then it will decay. The "Born" rule will spawn new cells adjacent to the clicked cell

The six tone generators work in L+R pairs, so there are actually 12 sets of parameters that can be affected by the cell values if the "Voices" is set to 6. Different parameters of the tone generators are shown to the right of the square display and by clicking on the row for each parameter, it will set the value for that parameter using the "Value" scaling to the bottom left of the square display. This then scales the range of possible values for each parameter which shows the offset / lowest value (left % number) and the range (right % number).

I thoroughly recommend reviewing [Misty Jones' YouTube videos](#) on this plugin. She has comprehensively explored and described the various options and settings.

So what this gives us is a way of generating tones. You can quantize changes to Hz or rhythmic pulses. In the screenshot below, I'm modulating the "Draw XY" left and right boxes and the "Value" box through the Strange Mod modulator (also by Dillon Bastan). This means that the Strange Mod modulation device is essentially drawing values into the square display. Outside the captured image, I'm also using another instance of Strange Mod to set values for some of the parameters.

The sound the generator creates is a little odd, but through use of additional audio effects it's possible to create some interesting evolving drone textures.



# 16 Tools - Creative use of audio effects

It *is possible* to create generative tracks that do not use reverb and delay. But to be honest, it isn't done often. In the preface to this book I said about ambient music: "Take any sound, drown it in reverb and delay... JOB DONE". You can, and I'd argue you *should* do better than that. Creative use of audio effects can accentuate harmonics and do interesting things with those; provide some grit, noise or warble; take a rather plain sound and use modulation and effects to twist it until it forms something much more interesting and evolving. All this BEFORE drowning it in reverb and delay.



## Key idea

Audio effects and automation together can take sounds and keep them interesting and evolving in ways that go beyond "static" application of reverb and delay on Return channels. Think of how you can use the huge range of creative effects provided by Ableton and employ these in ways that will take your original sound in completely new directions. This will provide the listener with some additional "ear candy" to keep the sounds interesting.

## 16.1 The tools of the trade

Ableton Live Suite comes with a HUGE range of audio effects and if you add in Max for Live effects there are almost too many audio-manglers out there to list or talk about in this chapter. While there are a few third-party audio effect plugins that are worth a mention, and which I use a LOT in my productions, you can get a LONG way with just the stock Ableton Live Suite audio effects. The benefit of third party plugins is where they offer something that the Live effects cannot. I'll talk about some of these below. But let's start looking at a few key plugins which are likely to be useful for generative music (and in particular for generative, ambient music which I create).

## 16.2 Hybrid Reverb

The [Hybrid Reverb](#) in Live Suite is a really nice reverb. It has a Convolution Impulse Response (IR) section which allows you to dial up real-world reverb from a wide variety of spaces and an Algorithm section which applies reverb processing and colouration. You can

blend between these or apply them in parallel and series. This in itself offers up a huge variety of sounds. This video will help explore what can be done with the Hybrid Reverb effect <https://www.youtube.com/watch?v=yLBIOiM97Vs>.

But when you dig into the Convolution IR section you'll also find options for textural processing - this applies a textural Impulse Response (IR) to the sound, so instead of a straight reverb tail, you'll get some additional texture from the IR audio file. The other thing to point out is that there is an option to have User-defined IR. This means that you can find IR audio files for different spaces online at sites like Open Air <https://www.openair.hosted.york.ac.uk> and use some additional IR files for some famous spaces, like Hamilton Mausoleum and York Minster. Obviously finding the right reverb sound for ambient music is kind of a key item. But you don't have to use just one. I like to use a long, lush reverb to help smear sound and create drones. I usually put this on a FX Return channel so I can route several tracks to it and save CPU. But you may also need a shorter reverb for other instruments to give them presence, in the way that a close mic to an instrument in one of these huge reverb spaces will capture more of the raw sound, while the gigantic reverb captures the instrument in the space itself.



## 16.3 Valhalla Supermassive

Here's a third-party effect I couldn't do without. [Valhalla Supermassive](#) is a free audio effect plugin from Valhalla DSP. It's a staple of ambient and drone producers because a) it's free and b) it has some MASSIVE reverb algorithms (duh!). Suffice to say it does exactly what it says on the tin - massive reverbs. But what Valhalla have done rather well is to combine algorithms and delay lines so that some of its algorithms behave more like delays, while some act more like reverbs. They also have algorithms that combine both - early delay turning into massive reverb. Some have short attack, some much longer. One of my personal favourites is the "Benson Arizona" preset which will turn ANYTHING into a massive drone. Combine this with Paul Stretch, as discussed in [Tools - Extreme audio stretching with PaulStretch](#) and you'll have instant drones.

To be honest, all of Valhalla DSP's plugins are excellent and reasonably priced. I particularly like the Shimmer device to add shimmer, stereo width and pitch shift reverb to sounds. Their

Delay plugin provides a wide variety of delay types, including tape delay (complete with wobble and drive) but also usefully a reverse delay (where the original audio is played backwards in the delay). I like the latter as the delayed line comes back transformed compared to the input. Again this adds some interest to the part.



## 16.4 Echo

I tend to prefer [Live Suite's Echo effect](#) rather than stock delay. There is a wide range of sounds possible using it, and some built in modulations and tweaks to the sound which add character. Use of Echo as a send effect has a long legacy in dub music where the DJ or producer would send a burst of signal to the Echo and by tweaking the feedback and EQ would create long tails of echo. You can learn more about the Echo effect here: <https://www.youtube.com/watch?v=4LxhIE169x4>.

Being a Lazy Producer, I prefer to let random processes tweak the Echo settings. I have built an audio effect rack with a trio of LFOs working on each other and then changing the feedback amount, lower and upper ranges of the EQ so that these wobble about randomly. Using an instance of Dillon Bastan's Strange Mod modulator will achieve similar effect as discussed in [Tools - Modulators](#). While the timing of the Echo is fixed, we have other parameters that mean that the sound is constantly shifting in an unpredictable way. This will prevent the Echo effect from being too "samey" across the track and provide some interesting moments. It's a bit like having a pair of "robot hands" that tweak the Echo settings.

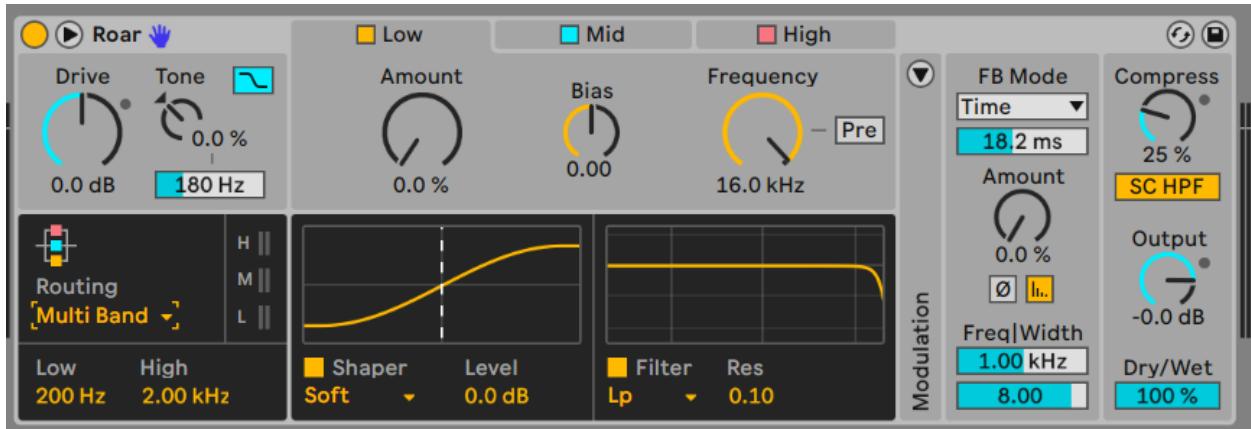


## 16.5 Saturation and Distortion

Saturation may seem like a weird choice for ambient music where much of the music produced is “pristine” and “delicate”. But there is a whole sub-genre where gritty, distorted sounds are most definitely a thing. My favourite from Live Suite is the [Saturator plugin](#) which provides a wide range of subtle distortion and drive. If you need more distortion than saturation then the [Pedal](#) plugin provides raw Overdrive, Distortion and Fuzz. Saturator allows you to drive the signal and dial in more subtle drive and overtones.



Live 12's [Roar](#) device takes these to the next level though and provides a comprehensive set of tools for applying anything from subtle warmth to the inputs to full-blown distortion. It has a wide range of routing options, including multi-band which allows you to dial in distortion for different frequency bands. This is a device that is really worth exploring to add texture in a wide variety of ways.



More about Roar here: <https://www.youtube.com/watch?v=ETzf6O9-6us>

## 16.6 Resonators

The [Resonator effect](#) takes an incoming signal and then resonates it by pitch-shifting additional resonators. You can dial-in the amount that the resonant frequencies are heard via gain adjustment, you can define the pitch of the resonator and whether the resonator is centred on a particular pitch. This can be useful to create drone noises and coupled with a long reverb like the Valhalla Supermassive you can create some really lush ambient beds.

I have found that a little drive to the input signal (which boosts upper resonant frequencies of the audio) works well with resonator to make the result more pronounced.



More about the Resonator effect here: <https://www.youtube.com/watch?v=wSjp6nX3rYI>

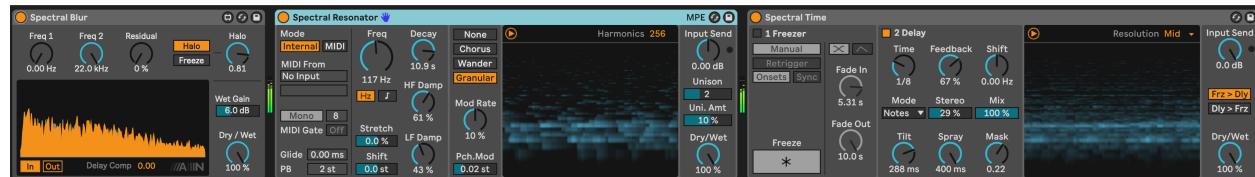
## 16.7 Spectral processing

Ableton Live Suite includes three Spectral audio effects. Spectral processing is interesting because instead of applying effects in a time-based manner (as audio comes through the effect) it is applying effects according to frequency (low to high) and loudness.

### 16.7.1 Ableton Spectral effects

**Spectral Time** which can be either apply a freezing algorithm to catch and hold incoming audio, or delay which applies delay and frequency shifting. The Tilt, Spray and Mask parameters apply the delayed audio to different parts of the frequency spectrum and gives some interesting smeared, granular and glitchy sounds. The Freezer and Delay algorithms can be used separately or chained serially from one into the other. Setting long fade in and fade out times allows you to catch and smear incoming audio which may be useful as part of a performance to join between two tracks or two sections of a mix.

**Spectral Resonator** is a resonator which focuses on different parts of the frequency spectrum, so you can tailor the resonance quite precisely. You can have a static tone for the resonance or pass in a MIDI part to change the resonance around a MIDI part. The different resonance algorithms - Chorus, Wander and Granular all have different sounds and it's worth playing with these to see which ones fit what you're trying to achieve.



Rishabh Rajan has useful YouTube videos explaining the Spectral Time (<https://www.youtube.com/watch?v=KUZLXAK8do4>) and Spectral Resonator ([https://www.youtube.com/watch?v=VAebi\\_brjEs](https://www.youtube.com/watch?v=VAebi_brjEs)) effects.

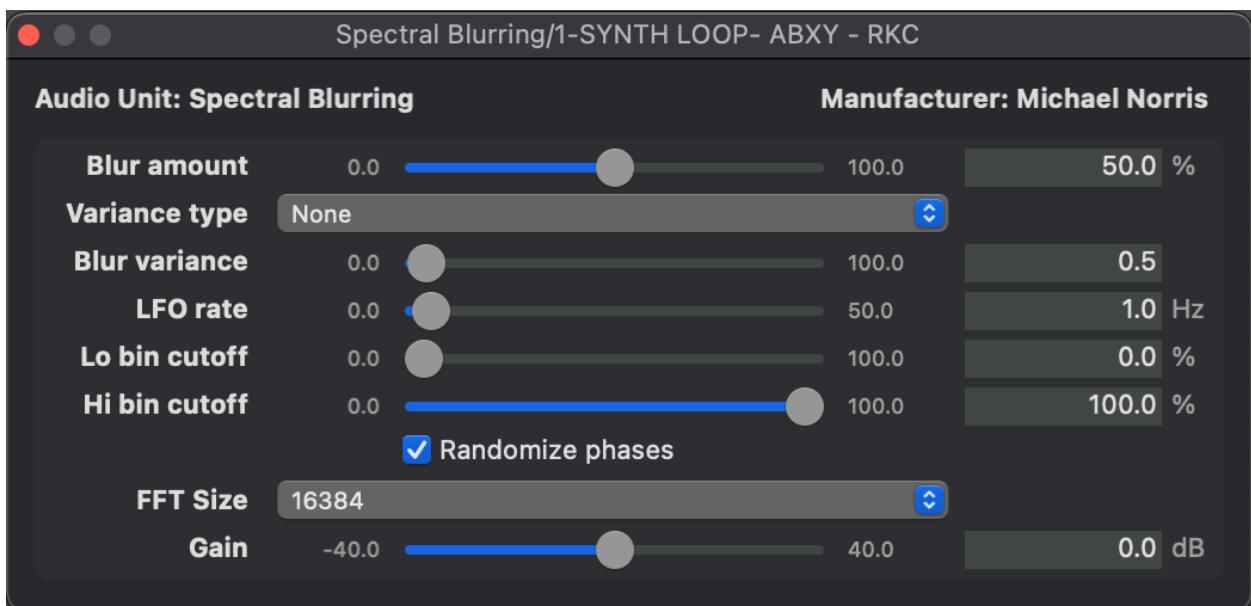
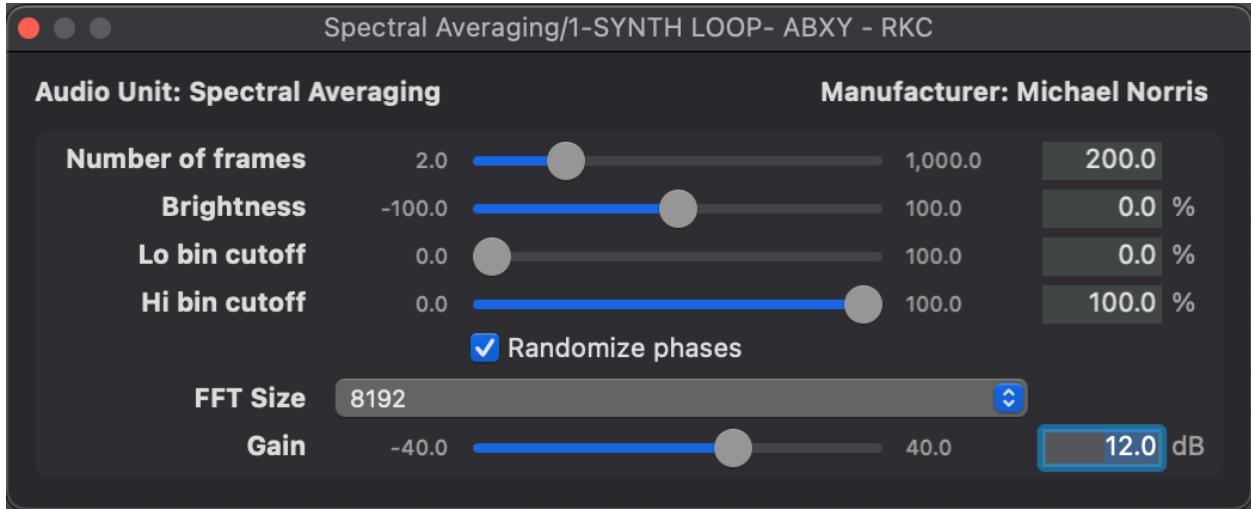
### 16.7.2 Michael Norris Soundmagic Spectral collection

Michael Norris has created some fantastic spectral plugins which can be downloaded here: <https://www.michaelnorris.info/software/soundmagic-spectral> as donation-ware.

The Spectral Averaging and Spectral Blurring are my two favourites from this collection to smear and blur incoming audio and create lush drones with the same sonic “fingerprint” as the original, but as a long drone texture. Applying these to the an audio clip or to a PaulStretch stretched audio will apply even more smearing and blurring to create fantastic drone textures.

The key parameter in both of these plugins is the FFT size which determines the size of the sample window in which the blurring or averaging is taking place. Larger FFT sizes smear

more, but take longer to fade in and out since the processing is happening over a larger window of audio. You may also find that you need to increase the gain of the output.



The whole collection is worth checking out for some very interesting effects, although as you can see above the user interface is VERY bare-bones.

## 16.8 Other notable mentions

### 16.8.1 Shifter

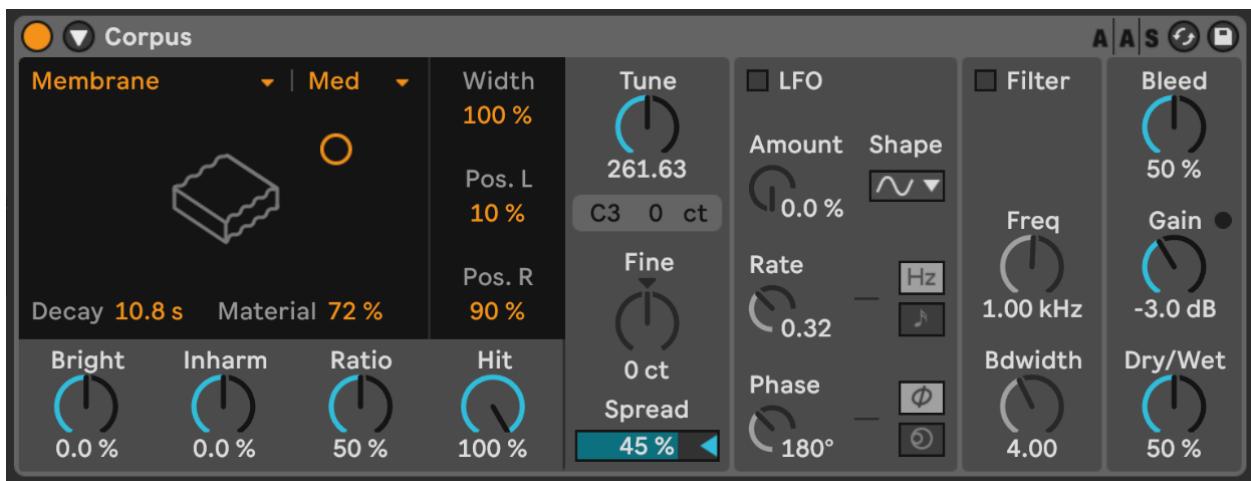
**Shifter** is a combination of a pitch and frequency shifter and ring modulator. It's a bit of a beast because it can be either subtle and also extremely weird depending on how much

shifting you do (and what type). This video provides a nice overview of the different modes and features: <https://www.youtube.com/watch?v=uqY8K8otbp0>



## 16.8.2 Corpus

Corpus is the audio effect companion to the Collision instrument. The input audio is used to excite a resonator - a plate, tube, membrane etc. - which then resonates. Depending on the resonator you choose you get different artefacts and results.



## 16.9 FX in series vs Effects Rack vs Return channels vs Separate tracks

There are so many routing options in Ableton Live that give us a HUGE range of possibilities when it comes to routing audio through effects. The effects can be on the track itself either in series (each effect passes its output on to the next effect) or in parallel using Effects Racks, on a separate track that brings audio from another track into it - with Monitor set to "In" and input selecting from another audio track or group bus - or in a Return channel where one or more tracks are routed into the Return send for processing. So which to choose?

I don't mind. It's YOUR track. But here are some thoughts on how *I* would use each option.

I would use effects in series where I want the effects to build on each other. This is helpful if you're going to use an effect to add some colour which can be exploited in the next effect, for example adding some grit that will ping a resonator, or a compressor to reduce the highs and bring up quieter moments which can then be fed into a saturator.

Effects in an Effects Rack are typically in parallel, although it's of course possible to build chains of effects within each chain or layer of the Effects Rack. I typically use these and then apply some modulation that will fade in and out each chain or select between them so that over time the different effects are not static over time.

Return channels are really useful if you'd like to send more than one track to an effect (reverb or delay) where you'd like to keep the effect consistent for each track or where you want to "mix" various tracks through an effect.

Using the output of one track as the input for another and then applying effects is a little like using a Return channel only the return channel is an ordinary Live track. This can be useful if you want to keep the "raw" track in the mix *and* apply effects separately, but it also allows you to pass the result of *that* track on into another track. So at any stage you can split out the audio, apply effects, route any of that signal to Return channels, then ingest the effected track into a further track for yet more processing.

So the choice is yours. Really. Try them all out. See which one works for you in a given situation. I don't mean that you should ***pick one*** and stick with it, but rather that you have a range of possible avenues to explore even within one track.

## 16.10 Use an Envelope with your effect

If your sound has a sharp attack, you may want to soften the send to the FX so that only the tail of the sound gets effected. Use an [Envelope Follower](#) effect before the audio effect. By setting the Map to 100% lowest and 0% highest (inverting the signal) it will duck the initial part of the sound (turning the send DOWN) and then bring it back up as the audio decays. Note that in the device below I have turned up gain to accentuate the behaviour (basically

to make the peak duck enough that the initial part of the audio has a low enough send amount) and I have smoothed out the Rise and Fall amounts which removes any “pumping” of the send to the Return channel effect. By setting a Delay you can also manually dial in how much or little of that initial part of the sound gets sent to the Return channel. This technique can be useful to capture and build drone sounds where you want to avoid sudden changes volume.



# 17 Tools - Extreme audio stretching with PaulStretch

[PaulXStretch](#) is a plugin that takes an audio file and stretches it to produce soundscapes and textures. This is not the type of stretching that is available in Ableton for warping audio by a few bpm to fit a different tempo, but instead is about sonic manipulation even to extreme amounts.

## 💡 Key idea

Taking an audio clip that has interesting harmonic structure, chords, textures and stretching it out using PaulXStretch can give you instant textures and drones. Even using short audio clips and stretching out 5x or 10x can give interesting textures to work from. You can also stretch by HUGE amounts - 100x for example - to smear something short and textured into something enormous. This is a nice technique to use when you have a short idea that has changing tonality and texture, and want to scale it up into something much more substantial. Applying PaulXStretch can spark new ideas...

## 17.1 Why would you want to stretch audio?

Stretching here means “smearing” audio, creating a blurred texture. The recognisable features of the original clip are no longer distinguishable, and instead you get something like a “ghost” of the original. The chords and tones from the original are stretched to an extent that they turn into drones and glacially shifting harmonics.

What this kind of stretching and smearing does is to give you a nice foundational element: a sound bed over which you can add other elements. Because the stretch smears out tones, it can sometimes add a slow movement of tonality rather than a sudden shift. This can be quite useful if you pick an audio source for stretching that works with the scale of the other processes in your track. It may for example allow you to imply a shift in tonality even if there's a repeating pattern (in a fixed scale) above it, much like how shifting bass notes below an ostinato line imply new harmonies.

## 17.2 What does it do?

PaulXStretch takes an audio file and can stretch it to a pretty arbitrary amount (maximum is over 1000x). This allows you to take very short audio clips as input and stretch them out to create soundscapes of pretty much any size. So if you take the 5 second Windows Startup sound, you can stretch this 60x to create a five minute soundscape.

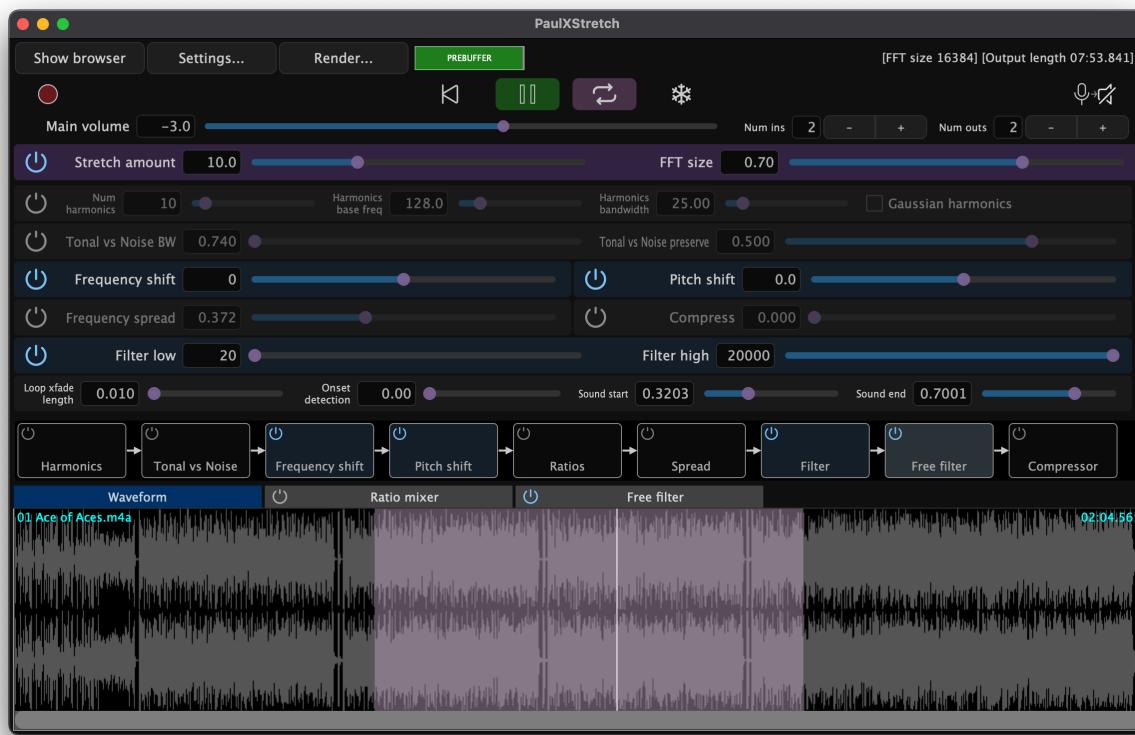
Some audio stretching algorithms, such as those used by default in Ableton to stretch audio, introduce artefacts and granulations once the algorithm goes beyond a modest decrease in beats per minute (bpm). But PaulXStretch also applies some (spectral) smoothing to create textures that are free of glitches - the sound is smeared evenly in the sonic palette.

The plugin also features various options to tweak the output of the stretched audio, including filters, the ability enhance resonant frequencies, pitch shift and more.

## 17.3 How do I use it?

You *can* use PaulXStretch as a plugin, but you can also use it as a standalone application. This would allow you to import, stretch, and then export the stretched file ready to import into your DAW as standalone audio. The latter is probably preferable in many cases - audition the file and stretching in the context of your tune, then commit and export to file.

## 17 Tools - Extreme audio stretching with PaulStretch



This video provides a longer explanation of the device and what it can do, including explaining many of the settings in the interface: <https://www.youtube.com/watch?v=oLvAbLEremk>

### 17.4 What should I use it for?

This is a plugin that favours some experimentation. You could take a short, textural clip that you like and then stretch this out to see how the textures react to extreme stretching. Note that if the input clip features some prominent harmonics or frequencies, these will be accentuated when the audio is stretched, so you may want to EQ the input clip **first** before applying stretch to eliminate any howling frequencies. I have had some success using short clips with interesting chord patterns, and then stretching this out to create a long-form piece. Because chord changes are preserved, but smeared and smoothed, the track then shifts very gradually in tonality, rather than being a long drone on a single tone, or rather than having sudden changes in tonality.

Not all audio clips will sound good stretched. My best results in experimentation have come when there are no drums or unpitched high frequencies (cymbals or high percussion).

As an experiment I once took the grindcore metal band Napalm Death's (in)famous "You Suffer" track, all 1.316 seconds of it (Guiness World Record holder for the shortest song), and stretched it out to 1 minute. The result was TERRIFYING. You can hear it here: <https://on.soundcloud.com/KXPPBRs1qdP6bSYt8>

A more sonically pleasing hyper stretch took the Windows 95 startup noise and stretched that out. <https://on.soundcloud.com/Q1fmUoNiyhA1BB3V9>

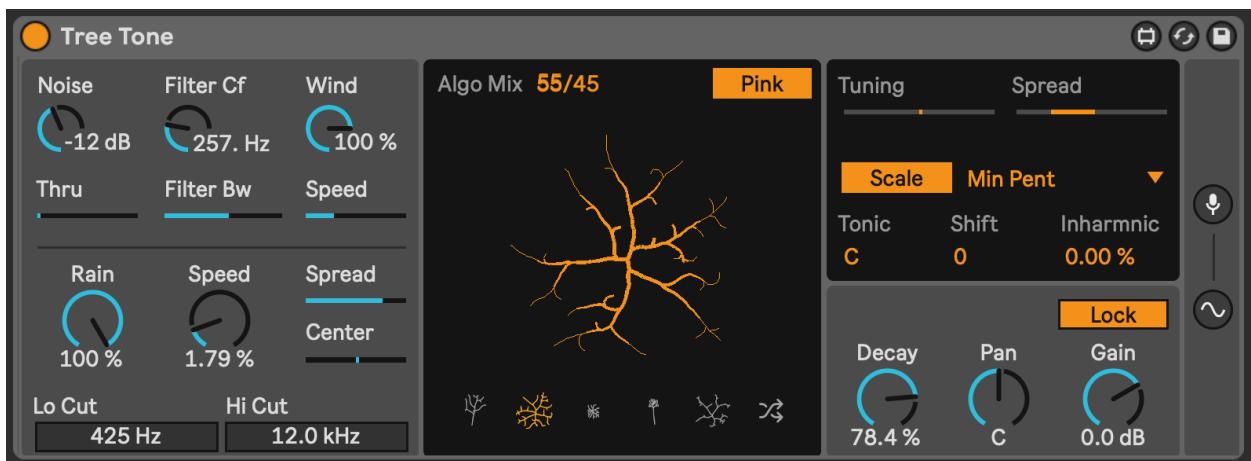
Another possibility is to use audio from other generative processes as the input to the stretching process. For example, try taking the process for Recipe - [Changing chords](#) but make the clip lengths short - say 3-9 beats long - and use that as the input to the stretching process. Since the chords are designed to change incrementally using that recipe, it should work quite well as the input to this stretching, smearing out the chord changes even more and making them implicit rather than explicit. Taking something random, then stretching and smearing is a way to create something potentially more sonically interesting.

I'm not guaranteeing that throwing any old nonsense through PaulXStretch will suddenly make it sound amazing, but it's sometimes worth a shot to see what the results will sound like. These can then be used as ambient "beds" or foundations for other sounds, textures and production on top.

# 18 Recipe - Making a drone using some audio generators and effects

In this recipe we'll make a simple drone track using a variety of audio generators.

1. Start by placing a Tree Tone audio generator on a track. I have chosen to ensure that the output is pitched and that it conforms to the C minor pentatonic scale. This will help integrate the sounds later.



2. On another track, add a Harmonic Generator (from the Drone Lab pack). This is an instrument combining sine-wave drones oscillating at fixed pitches relative to the Root note. We can experiment later with this to see how changes to pitches or FM Ratio change the resulting sound, and maybe automate these via LFO, but for now, let's keep the drone from this to be quite static.



## 18 Recipe - Making a drone using some audio generators and effects

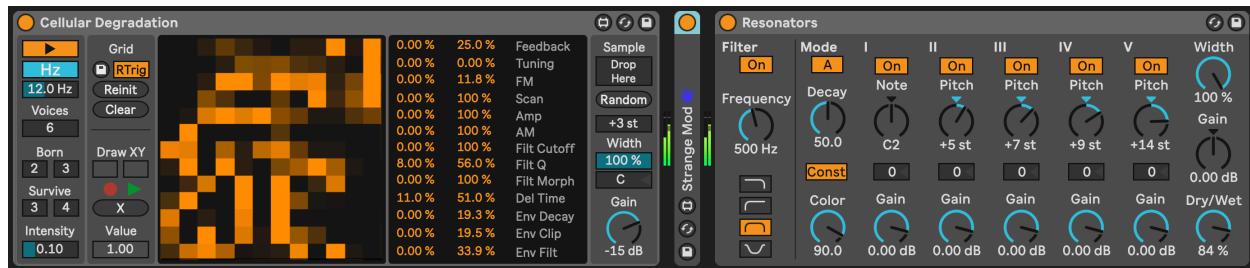
3. Next let's add a real odd-ball: the Cellular Degradation plugin by Dillon Bastan. As described in [Tools - Audio Generators](#), this is a weird instrument combining 5 different stereo FM oscillators, and it can make some VERY weird noises. That's what we're going to do here. Normally you "draw" into the main panel to make cells "alive" and then the cellular birth and death process retains these or allows them to "die". To automate that process we're using another favourite Dillon Bastan plugin, the Strange Mod modulation device. We can then use its chaotic modulation processes to automate "drawing" the X and Y values. This should produce some very strange buzzy textures.



4. At this point I added a simple bass root note drone (on C1) using the Drift instrument. This helps anchor our drone.

Since we want to balance the unexpected and predictable, here the bass drone note, and Harmonic Generator provide the "anchor" around which we can place Tree Tone (as a mildly chaotic process) and the Cellular Degradation as a VERY chaotic process.

5. From here, feel free to add effects on Return channels (Reverb and Delay are good options) and / or apply resonant processing within the individual tracks - like adding the Resonator or Spectral Resonator plugins. Experiment. See what works. I found that putting a resonator on the Cellular Degradation track and tuning additional resonance pitches within the C minor pentatonic scale helped to "tame" the weirdness, if that's what you'd prefer. Another decent option (in addition or instead) might be to drive the Cellular Degradation track using a Saturator or distortion.



6. Sometimes, if using an audio generator it helps to add a compressor (occasionally a multiband compressor). This will reduce the peaks and can also bring up some of the quieter moments (if you increase the make up gain). If you're using additional processing like a resonator, this is also helpful since it may "ping" the resonator with additional sounds, while avoiding very loud moments that can catch the listeners ear.

*18 Recipe - Making a drone using some audio generators and effects*

7. Press play. Sit back and relax.
- 

You can hear my version of this track here: [https://soundcloud.com/mikeksmith/5\\_audio\\_generator\\_drone/s-Nw7nwUKDhTg](https://soundcloud.com/mikeksmith/5_audio_generator_drone/s-Nw7nwUKDhTg)

# 19 Process - Using Scales

As discussed in [Tools - MIDI tools](#), scale quantisation is a widely used tool to harness the random note values from MIDI Generators. It provides *just enough* quantising to turn something unlistenable into something that's still random, but perhaps more listenable. The problem is that even within a defined scale, there are sometimes notes that will irritate the listener if they happen too often - for example in the C Major scale, the note B - one semitone below or 11 semitones above the root is always going to be a bit jarring. In the context of most Western music that note is often played in passing on the way back to the root note C. But in the context of random note generators it may pop up a little too often for comfort and not always resolve to the C. So what do we do? Avoid Major scales?

## 💡 Key idea

Finding a few useful scales that work well in a generative context is a good idea. This is largely a matter of taste, and potentially just trial and error to see what you like, but there are a few guiding principles that might help inform your choice. Choosing scales or eliminating *awkward* notes from a scale can also help make music where the tonal centre of the music isn't locked in or *obvious*, which allows you to imply a change of key through a root note change, without having to change the scale for the generated (or quantised) MIDI. I'll call this being tonally ambiguous.

## 19.1 Pentatonics - Minor and Major

You know how you can play only the black keys on the keyboard and no matter what you play it just all somehow "fits" and sounds good? That's because you are playing a pentatonic scale. Many guitarists will scoff at players who riff away on a pentatonic scale, because it is seen as being somehow a "cheat" and avoids having to learn more complex scales, modes and harmonies. But for our generative purposes, ***pentatonic scales are your absolute number 1 pick.*** You can combine any number of generative processes working within a pentatonic scale and they will fit together like hand in glove.

Below are the notes and scale tones of the C pentatonic scales:

C min pentatonic	C	Eb	F	F	Bb
Scale tones	1	m3	4	5	b7

## 19 Process - Using Scales

C Maj pentatonic	C	D	E	G	A
Scale tones	1	2	3	5	6

C (neutral) pentatonic	C	D	F	G	A
Scale tones	1	2	4	5	6

If you pick any 3 out of either of these scales then you get nice triads from their respective (non-pentatonic) scales.

## 19.2 Hirajoshi

The Hirajoshi scale is a scale derived from Japanese shamisen music and the tuning of the koto instrument. It is a pentatonic scale, but with some interesting properties that distinguish it from the more frequently used versions above. It has the minor 3rd in it, which gives it a slightly melancholic feel, but also the minor 6th - C - G♯ (opening two notes from the theme from Star Trek).

C Hirajoshi	C	D	D♯	G	G♯
Scale tones	1	2	m3	5	m6

## 19.3 Modes

Modal scales are **hard** to wrap your brain around at first. In practice, they are very easy. Take the notes in the scale of C, start on different note and end on the note one octave above that.

### 19.3.1 Mixolydian mode

Start on the G of the C major scale and play only the “white keys” of the C major scale up to the next G. That’s the Mixolydian mode. Let’s look at the notes and scale tones and try to understand what’s going on though:

G Mixolydian	G	A	B	C	D	E	F
Scale tones	1	2	3	4	5	6	b7
C Mixolydian	C	D	E	F	G	A	Bb

Mixolydian is essentially a major scale but with a flattened 7th. This is a nicer alternative to the straight major scale because the harsh major 7th is “softened” by flattening it. You can combine this flattened 7th with other notes in the scale and get attractive chords. You can also swap between the Mixolydian mode and the fourth of the natural major key. So C Mixolydian has the same notes as F Major. G Mixolydian has the same notes as C Major. If your MIDI note generator process pings out the flattened seventh here, it won’t sound harsh and abrasive. And you could either play a root drone of the Mixolydian mode root, or the fourth of the natural key and both would sound fine.

### 19.3.2 Dorian mode

D Dorian	D	E	F	G	A	B	C
Scale tones	1	2	b3	4	5	6	b7
C Dorian	C	D	Eb	F	G	A	Bb

Dorian is a more “minor” sounding mode (with a minor third from the natural scale) and also the flattened 7th we talked about above. Dorian is essentially the “minor” version of what we discussed above with the Mixolydian, and it works for many of the same reasons.

## 19.4 There are no “wrong” notes, only context

To paraphrase an old saying “Play a note once and it might be wrong. Play it twice and it’s jazz...”. There is certainly a truth that there are truly no “wrong notes” - whatever note you play you can add additional notes around or resolve it to a point that it will work within the context. When you compose or play with others, you can work on the context or resolve these problematic notes towards something more comfortable. This tension and release is a great feature of many tunes. But within generative music, we can’t guarantee that the processes we have set up will provide that context or resolve the notes towards something more pleasant.

If you are using generative processes to create something to play along with - where *you* can provide the context and resolution, then go ahead and try out other scales. In fact, this might be a really good musical challenge... But this is a little beyond the scope of this book. And my skill set.

## 19.5 Selections of notes - chords

In [Recipe - Changing chords](#) and [Process - Note-wise probability](#) I talked about stacking notes in a chord and then applying probability. I mentioned that you might want to choose notes that would work well together, so that regardless of which notes were chosen you would

get something that didn't sound too "weird". Now given the context of the scales above, I hope it will help point you towards scale tones that would work over a selection of root notes, or where a random selection of 3 or 4 (or 5 or 6!) of the notes would produce something that would still be a "nice chord". Also contextualising this collection of notes (chord!) with a given root note or bass note for the chord would imply one key / mode or another.

## **19.6 Other scales & traditions**

In Western music we have a view of what collections of notes and pitches sound "nice" or "harmonious" and, to our ears, what "works" and what "doesn't". Of course, there is a huge world of scales in the world across a huge range of musical traditions. In this chapter I've only focussed on some example scales. With Ableton's Scale plugins and scale awareness you might well find others where generative methods would give interesting results. Within the DAW, experimenting with these scales is only a click away...

# 20 Process - Using Field Recordings

Computer goes “Bleep bloop bleep bleep bloopy bloop. Whoosh. Bleep bleep... BRRRRR....”

If the sound sources for your generative music using synths and sound generators that don't involve sounds from “real world” instruments and you add in random and probabilistic processes then pretty quickly you have moved away from anything that could be described as “natural” sounding. How can this kind of music appeal to humans, who by their nature are more used to organic sounds and textures they hear day to day?

 Key idea

Field recordings of nature, or sounds that humans might hear day to day add back in context that might help listeners feel more engaged with the generative and more random aspects of the music. The sound of a stream, birds, rain, wind, people talking in a street and cars passing, even the hum or mechanical clicks of machinery can help the listener create a mental picture to go with the more abstract or generative ideas.

## 20.1 Balancing the natural and the (unnatural) randomness

Field recordings of natural, human sounds provide that balance between something familiar, predictable and natural with random bleeps and bleeps. In [Process - Balancing the unexpected and the predictable](#) we discussed providing musical balance to the randomness. In this chapter we're talking about using field recordings to provide that balance.

## 20.2 Natural and urban sounds

Take a walk in a woodland area. It's never entirely silent. There's likely the sound of wildlife, wind in the trees, perhaps the sound of water. By describing it to you now, I'm sure you have a mental picture (even though it's sound!) of what you might expect to hear. Similarly, you can use urban sounds (street sounds), sounds from the house, mechanical noises - basically anything that listeners might be familiar with.

## 20.3 Randomness does occur in “natural” sounds

The sound of a wind-chime is often thought of as soothing. The wind gently causing the chimes to sound, little waves of air that cause the chimes to gently ripple, with the chimes tuned to pitches that work well together... Sounds a lot like some of the techniques we've been talking about, right?

The sound of birds singing in the forest, with the odd chirp of a squirrel or the steady rasp of a frog. Nobody is telling them when to make sounds. It's all pretty much random. So why (again!) is it that our brains think of these sounds as soothing?

I think the trick here is that we have context for the randomness. When we hear those sounds we know how they occur and where. But this underlines (at least for me) how adding context or something relateable to our compositions can help the listener engage better.

## 20.4 Recording noises

You likely have a phone with you now, or perhaps nearby. In that case you have a microphone to hand. So next time you're out for a walk, take out the phone and record some noises. All you need to do is to be ready to listen and decide how the sounds you're hearing are interesting, or how they could be contextualised (or provide context). The fidelity of these recordings doesn't have to be astounding, after all, they are likely to be low in the mix. But minimising wind noise or handling sounds if you can will help the later “clean up” and integration of these sounds into the music. Use of EQ to remove low end or high end noises might be necessary.

## 20.5 Online sources for field recordings

Many producers create field recording sound packs, and some of these are sold through places like Bandcamp. Searching for “Field Recordings” should allow you to find these. Please do check licensing terms and conditions before you reuse these in your own compositions though!

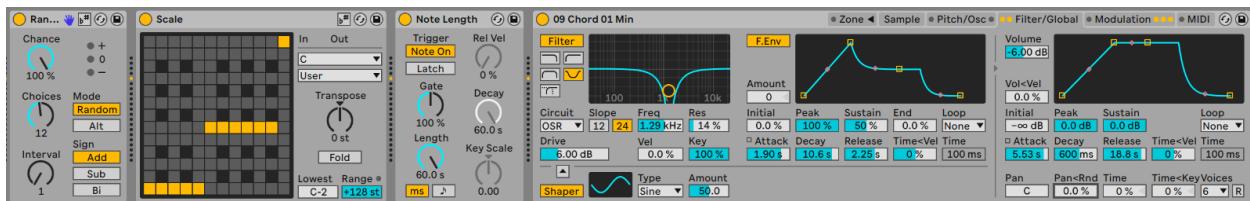
## 20.6 Using the samples

You can use these as is, alternatively you might want to put the sample into Ableton's Simpler and have a MIDI note trigger the sample but with random modulation on the sample start point. This will vary the starting point of the sample with each MIDI note and allow you to effectively extend the length of the sample to an arbitrary extent.

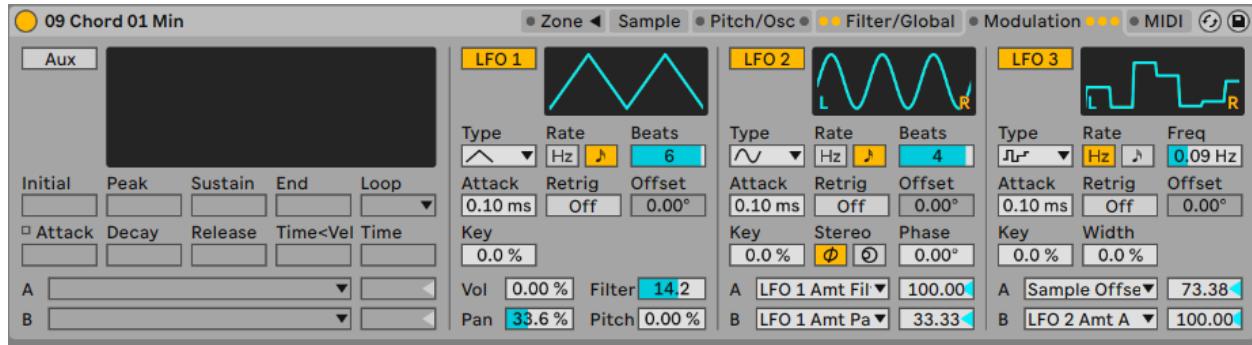
# 21 Recipe - Audio clip or Sampler into effects

The Sampler instrument from Ableton Live is often used to slice and dice breaks and loops, or to produce multi-sampled instruments that allow you to recreate sounds and playback across the keyboard. But in our context we're going to use it to simply play back a longer textured sound sample and apply some modulation to the cutoff, producing a drone sound. We can then throw this through a barrage of effects - either using Return channels or via individual audio effects with the sampler output routed into each. Effectively we're reusing an idea we've used before, but starting with Sampler as the audio generator. You could alternatively use the audio output from [PaulStretch](#) as the drone generator. The benefit of using Sampler is that we can add MIDI effects to (randomly) change the pitch of the sample so that it doesn't have to be consistent from one loop to another. Other sample-based instruments like the Max for Live Granulator will allow you to do similar.

1. Create a track with the Sampler instrument and one clip with a single note of 8 or 16 bars.
2. In the Sampler Filter/Global section, give the envelope a long attack (several seconds) and very long release (again, several seconds). This will allow different notes to blend rather than cut each other off.
3. In the Sample Modulation section, create some modulation on Filter, Sample Offset and other parameters so that the sounds morph and change over time rather than staying static. You can also modulate the LFO amounts or rates in LFO2 and LFO3 so that the modulation itself isn't static. Modulating the Sample Offset makes Sample work a little like a granulator - instead of starting at the beginning of the sample each time, the modulation picks a new position to start from. This prevents the sound from being too predictable with each new note.



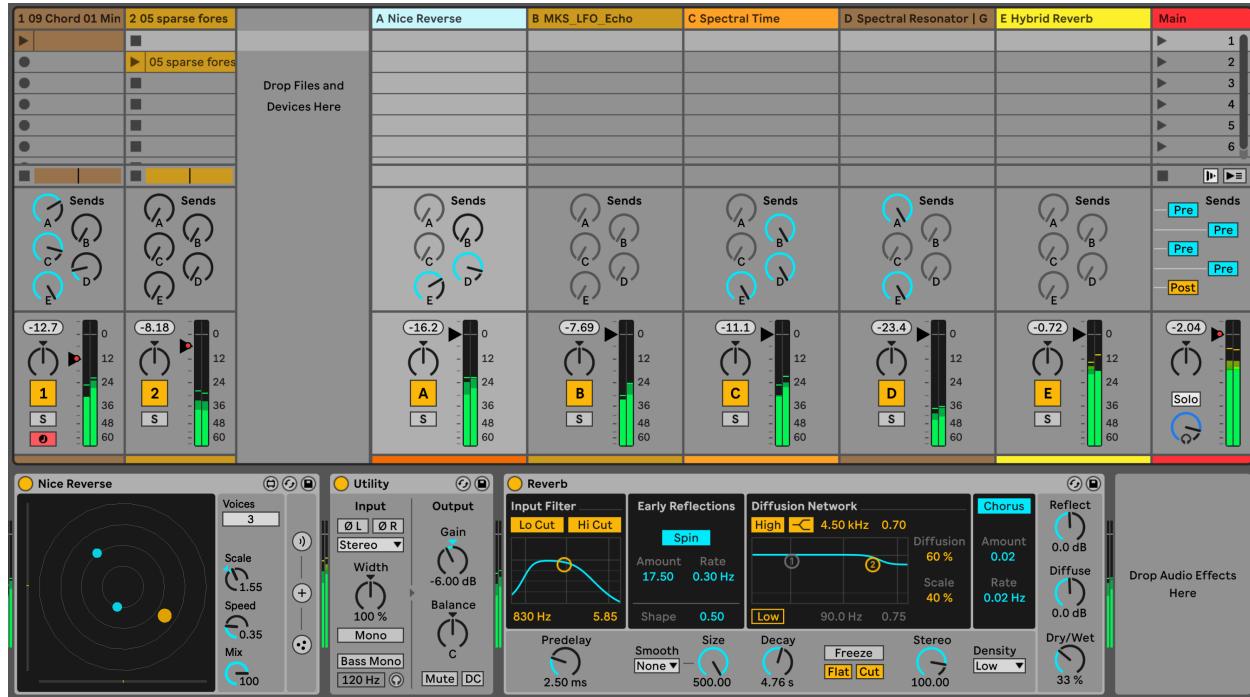
## 21 Recipe - Audio clip or Sampler into effects



4. Using techniques described in [Tools - Creative use of audio effects](#) and used in the [Recipe - Making a drone using some audio generators and effects](#) set up a network of audio effects on effect sends / Return channels.

In the example below the Sampler instrument is in Channel 1 and is sending audio out to a variety of FX channels. Note that the Return channels are also routing audio around between them to further add layers of effects. You need to be a little cautious when doing this to prevent feeding audio back into an FX channel and creating a wall of harsh sound. You can see that Return A (the “Nice Reverse + Reverb” effect) is sending audio out to Return D (Spectral Resonator) and that that channel is sending audio back to Return A. This is a little dangerous, but being cautious with the send amounts and watching for signal overloading, it is possible to get away with this. Adding a Limiter to the Return or Main channels would prevent runaway sound feedback causing physical damage to ears or speakers.

Spectral Time and Spectral Resonators are interesting effects to add texture and resonance into fairly static inputs (like our drone here).



## *21 Recipe - Audio clip or Sampler into effects*

5. You can apply MIDI effects to the Sampler track to pick a new random & quantised note each time the clip restarts. Unlike using an audio clip, using Sampler allows us to repitch the sound with each new note.
  6. Once you have a setup that works and produces pleasing sounds, try dropping a different pad sample into the Sampler. Doing this allows you to change up the sounds generated, as each sample will have different tones, resonances and features.
  7. Press play. Sit back and relax.
- 

You can hear my version of this track here: [https://soundcloud.com/mikeksmith/6\\_sampler\\_drone/s-sDz7Ts85Fs9](https://soundcloud.com/mikeksmith/6_sampler_drone/s-sDz7Ts85Fs9)

# 22 Process - Unsynced loops

“Around the world” - Daft Punk

Everyone loves a good loop. If you get a good idea of a pattern or something that lasts 8 bars then you can let that good thing just loop around and around and around (the world). A large part of The Lazy Producer’s arsenal depends on loops and repetition - after all we’re balancing predictable and unexpected, right? But if there’s something in your brain right now that’s saying “No, 8 bar loops repeated infinitely really *is* the ultimate Lazy Producing...” then I’d have to agree with you. So how do we break away from the tyranny of the 8 bar loop? Arrangements? Verse-Chorus-Middle 8 structures?

Before you abandon the ideas of generative music (which doesn’t really lend itself well to arrangements, verses or choruses), there’s a trick up our sleeve which is to take our loops and make one of them shorter or longer than the other.

## 💡 Key idea

Making loops different lengths combines the predictable: each loop contains the same music idea each time - with the unexpected: the combination of those musical ideas each time the loop cycles creates something new and unheard.

## 22.1 Set phases to stunning

Steve Reich and other minimalist composers worked with musical phrases that each performer played a number of times. Sometimes the number of repetitions was at the discretion of the performer, sometimes governed by the number of times the performer could play it in one breath, sometimes the composer worked with actual tape loops of different lengths. The concept is that while each part is repeated, the repeated parts move out of phase with each other and so create new sounds, rhythms and harmonies. No two performances would be the same, since the number of repetitions isn’t fixed.

## 22.2 Easy implementation in the DAW

Creating loops within the DAW is easy. Almost *too* easy. But often our “default” is to make those loops the same length. So the same ideas cycle round. But if we make the loops

unequal lengths then it'll take longer for those ideas to sync up. And if the loop lengths are not even - break out of the 2, 4, 8, 16 bar patterns - then it'll take longer for them to repeat.

We actually used loops of different lengths in [Recipe - Changing chords](#).

## 22.3 I'm in my prime

If you remember back to your basic maths from school, you may remember that you can break down most numbers into their prime components: 2, 3, 5, 7, 11, 13, 17, 19 etc. What this means for us as Lazy Producers is that if your loop is a prime number of beats / bars long then it'll take MUCH longer for those loops to sync back up. If you have a 4 bar loop and an 8 bar loop then it only takes two repetitions of the 4 bar loop before you hear the same music idea again. If you have a 3 bar loop and a 5 bar loop then it'll take 15 repetitions before you hear the same idea again. This means then that even for two musical ideas we get a lot of variety when you combine the loops even though both loops are repeating fairly often and fairly quickly. When you combine this idea across more than two loops you can get huge amounts of varying material with very little chance of repetition.

## 22.4 Fibonacci and the Golden Ratio

Primes are nice for preventing repetition in the grand scheme of things, but there are other options. Again, back to early maths and you might remember Fibonacci sequence where each number is the sum of the previous two numbers, starting with 0 and 1. So the first ten numbers are: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34. Notice that there are some prime numbers in there (2, 3, 5, 13) but also some even numbers (8, 34).

If you take the Fibonacci sequence and layout squares with their sides the length of the Fibonacci number, you get what is known as the “Golden Ratio” where the sizes of the squares grow in relation to the ratio of the previous two parts. There is a familiar pattern also if you connect arcs - quarter of a circle with radius equal to the size of the square. We often see this pattern in nature - in the pattern of seeds or petals on a plant.

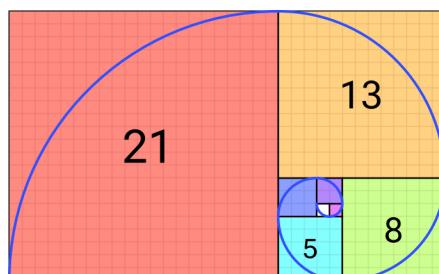


Figure 22.1: By Romain - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=114415511>

I'm not going to pretend that it has mystical properties (it's only maths after all) but the combination of primes, even and multiples in the Fibonacci sequence might be the ultimate sweet spot of combining things that occasionally repeat and things that take a long time to repeat.

Suffice to say, it's worth playing with some of these ideas to break out of the predictable loop and make something that perhaps could retain interest for a long period. In our next Recipe section we're going to tap into this idea.

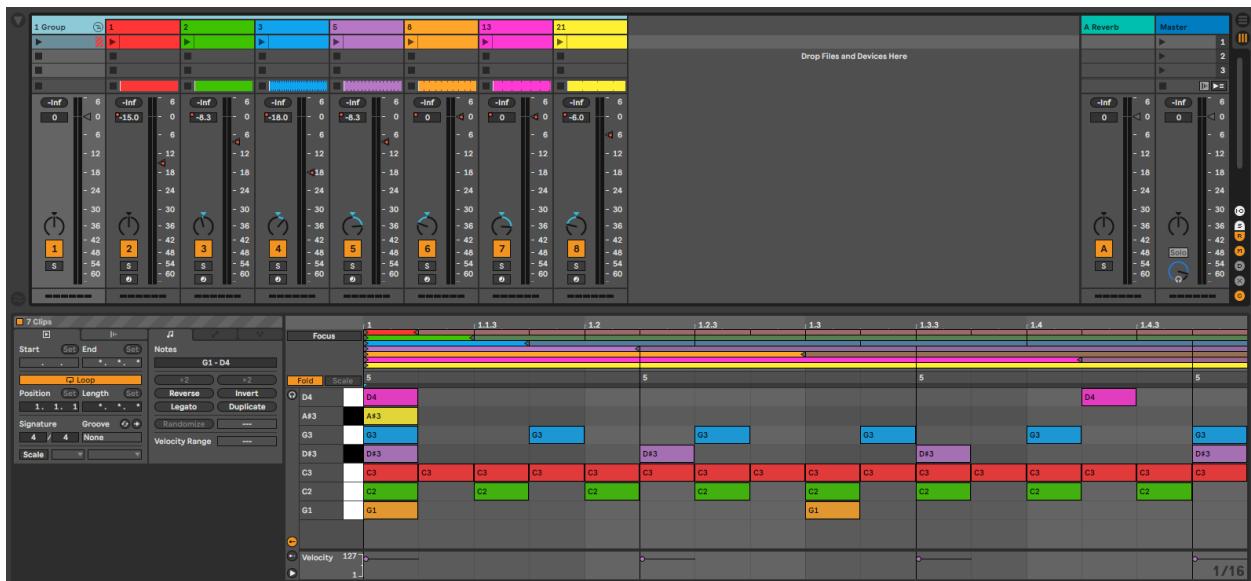
## 22.5 Let's ROTATE

In discussion of Euclidean Sequencers in [Tools - MIDI Generators](#) we talked about how you can rotate a sequence so that it doesn't start at 12 o'clock position (or on the first beat of the sequence). The same is true of loops. To break out of the predictable sequence you could choose to jump the loop start point to a different position. Sometimes small things like this will unveil new ideas, new rhythms which may be sufficient to keep interest.

# 23 Recipe - Using the Fibonacci sequence to create loops (after DivKid)

This Recipe comes from DivKid (Ben Wilson) who shared it on his [Instagram](#) channel. I'm very grateful to Ben for giving me permission to share this idea in this Recipe. I would encourage you to check out DivKid's content (<https://www.youtube.com/divkidvideo>) - he has a lot of really great tutorials, walkthroughs and ideas for use with modular synth rigs, but which can also be applied to Ableton Live. You can also sign up to his [Patreon](#) where if you sign up as a Patron you can download a copy of the Ableton set that goes with this Recipe.

1. Create one loop per track in an Ableton Live set according to the Fibonacci sequence length. Initially, set up the loops where 1 = 1/16th Note, so 2 = 2/16ths, etc. You can experiment with longer loop lengths where 1 = 1/8th Note, but obviously these will repeat much more slowly.
2. You can choose the notes and sounds you wish to use for each track / loop. You might like to use a combination of kit sounds and melodic sounds. You can choose notes from a scale or from a chord e.g. C, Eb, G, Bb, D.



## 23 Recipe - Using the Fibonacci sequence to create loops (after DivKid)

3. You can add modulation for the sounds, add randomness in velocity of the notes to prevent them from becoming too “samey” or you could add the occasional random / quantised pitch shift.
4. In his example set, DivKid has used plucky Operator with Sine tones but with additional tone shaping, sound design and granular delay to the grouped tracks which breathes life into the results and prevents them from becoming too stale.



5. Given this basic set up you could apply many of the concepts and ideas presented in earlier chapters to further use MIDI in other instruments e.g. pads, or take the resulting audio and process it further.

6. Press play. Sit back and relax.

Here is DivKid's version of this recipe: <https://www.instagram.com/reel/C2ID25WqB0l/>

# 24 Tools - Live 12 Overview: MIDI and compositional tools

During the process of writing this book, Ableton released Live 12. With Live 12 there are a whole host of improvements and additions which make the life of someone making generative music a whole lot easier. In this chapter we'll look at the new compositional and MIDI tools. In the next chapter we'll look at new sequencer tools. In subsequent chapters we'll look at updates to modulation, and some specific Transformer and Generator tools.

## 💡 Key idea

Live 12 contains a number of tools and new features which enhance how we can make generative music using Ableton Live. Some of these are enhancements to existing functionality, but there are many places where there are brand new features which could be used to create generative music, or provide idea generators for composition.

## 24.1 Scale awareness

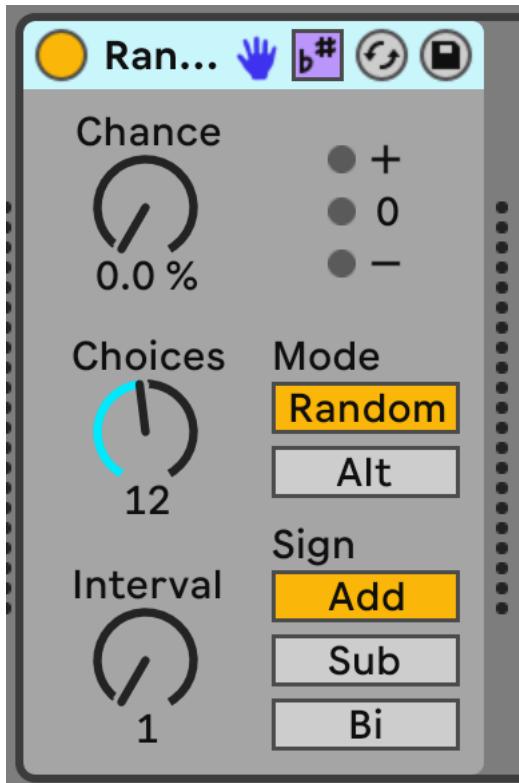
Live 12 now has a feature where certain MIDI devices, audio effects and instruments are “Scale Aware” so that when you dial in pitch related changes, these will respect the current scale for the Set. The “Current Scale” is set either when you create a new Live Project, in which case new clips inherit the current scale setting, or the clips in a Scene can define the current scale for that scene. For individual song projects, this works very well. If you have a combination of scales defined across a Project - for example in live performance - it pays to be a little careful to ensure that MIDI clips within scenes do have scales defined in order to change the Current Scale setting between scenes. Proceed with a little caution in live performance is my advice!

The purple b# button means that the Current Scale setting is being honoured by the MIDI effect, audio effect or instrument.

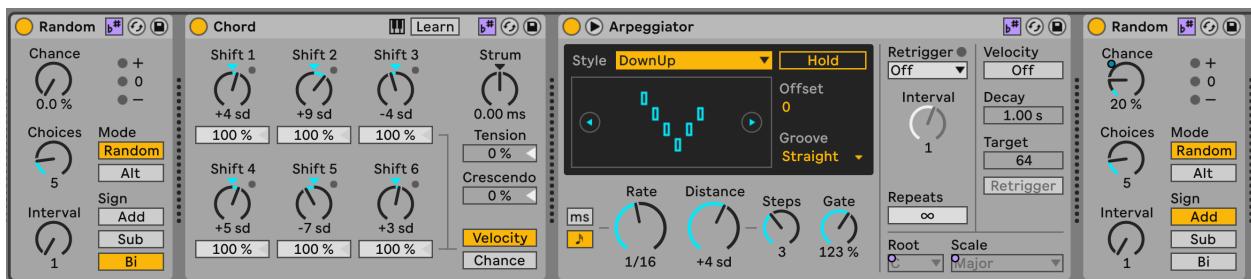


For more on Current Scale settings see <https://help.ableton.com/hc/en-us/articles/11425083250972-Keys-and-Scales-in-Live-12-FAQ>

In the “Random” MIDI device below, the purple “Scale Mode” or Scale Awareness button is lit, meaning that random pitch shift will be in the context of the Current Scale setting, so shifting up or down scale degrees, rather than semitones. This is a boon for the Lazy Producer because now you won’t need to add MIDI Scale quantizing devices, and also changing the scale in the Current Scale (or through clips), the MIDI devices and instruments will automatically pick up whatever the new Current Scale is.



So now we can chain together MIDI Devices as discussed in [Tools - MIDI tools](#) and used in [Recipe - Doing more with less using MIDI tools and plugins](#) and by turning on Scale Awareness we'll pick up whatever the Current Scale setting is. Note in the Chord and Arpeggiator devices that instead of “st” semitones for shifting, it is now labelled “sd” scale degrees. This takes a little getting used to because the scale degrees are added onto the MIDI note value - so +4sd is a fifth above the incoming MIDI note.



Scale awareness also extends to some of the Synth instruments in Live 12 - Meld in particular has scale awareness for some of its tone generators, where changing the macro knob for that generator adds harmonic content that is in tune with the Global Scale.

🔥 Continue to use Scale Device in live performance!

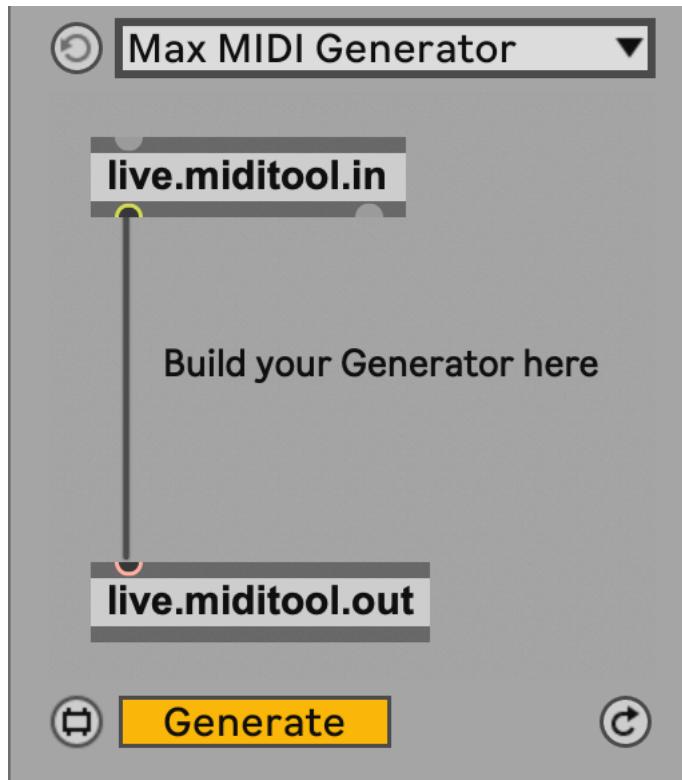
I would continue to use the Scale MIDI Device for live performance just as a fall back to be certain sure that you don't unexpectedly have one clip that's playing G minor, while the Current Scale thinks its C major.

## 24.2 Compositional tools

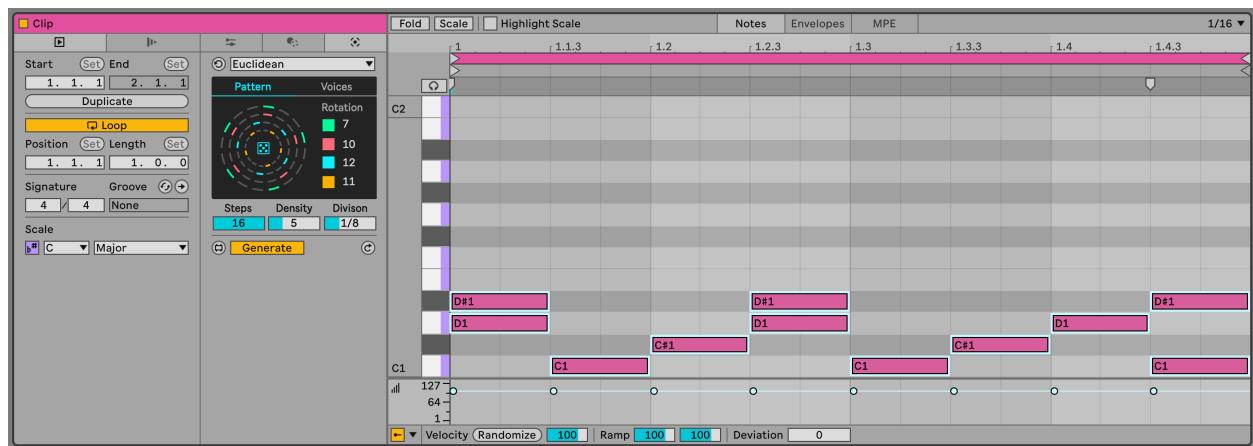
### 24.2.1 MIDI Generation

Live 12 now comes with two new features built into MIDI clip creation: MIDI Generation and MIDI Transformation. MIDI Generation does what the name suggests - provides a range of different methods for generating new MIDI parts: rhythm - which creates sequences of MIDI notes by specifying the density of notes within a certain number of steps and then allowing you to choose permutations of these, stacks - which defines chords, seeds - which generates one or more "voices" of completely random notes (but these could conform to the clip scale setting) within the clip, shape - which generates notes by drawing in a pattern or shape, and Euclidean generators. And because what sits behind these generators is the power of Max for Live, the possibilities open up for a huge range of innovative MIDI generation.





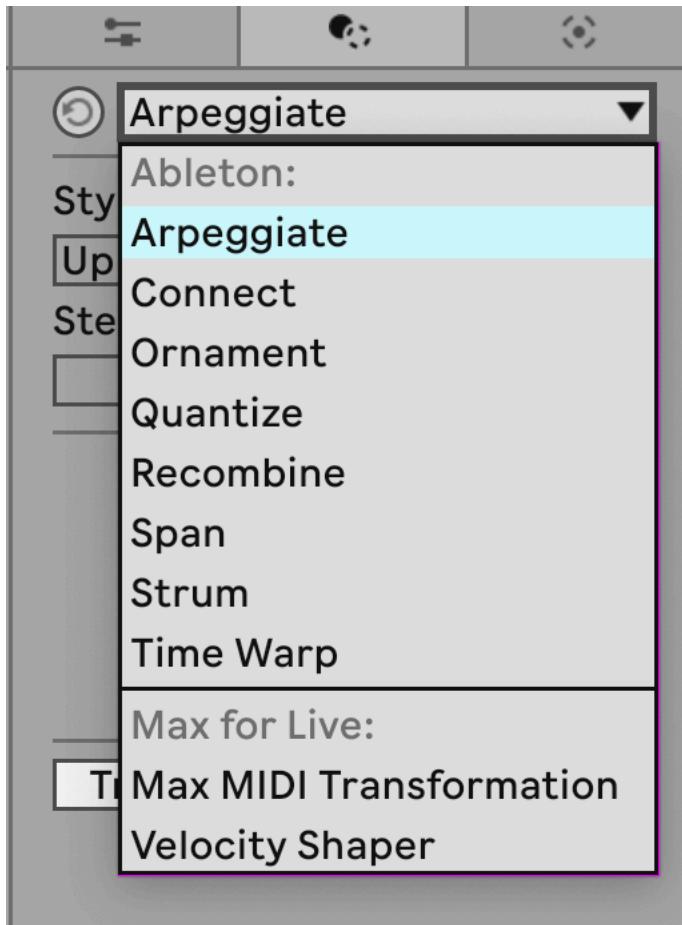
In Chapter 13 we looked at MIDI Generator devices including Max for Live devices. But having some similar functionality here “baked into” the Live MIDI clip tools puts these tools at the fingertips of anyone with Live 12 Suite.



## 24.2.2 MIDI Transformation

The MIDI Transformation tools includes tools to arpeggiate the notes within the clip, connect - connecting existing notes in the clip with passing tones, ornament - adding additional short notes before the main note, like a flam in drum terms, or mordent in classical notation, quantize - here applying quantisation non-destructively, i.e. you can see the effect of the amount of quantisation on the MIDI notes, recombine - which lets you select notes and then

permute these (rotate the order, mirror, shuffle), span - which allows you to change the note lengths by various means, strum - which applies a rhythmic strum pattern to chords, and time warp - which can bend the timing within a bar, so although the bpm clock remains constant, MIDI notes in a clip can trigger with a more fluid timing.

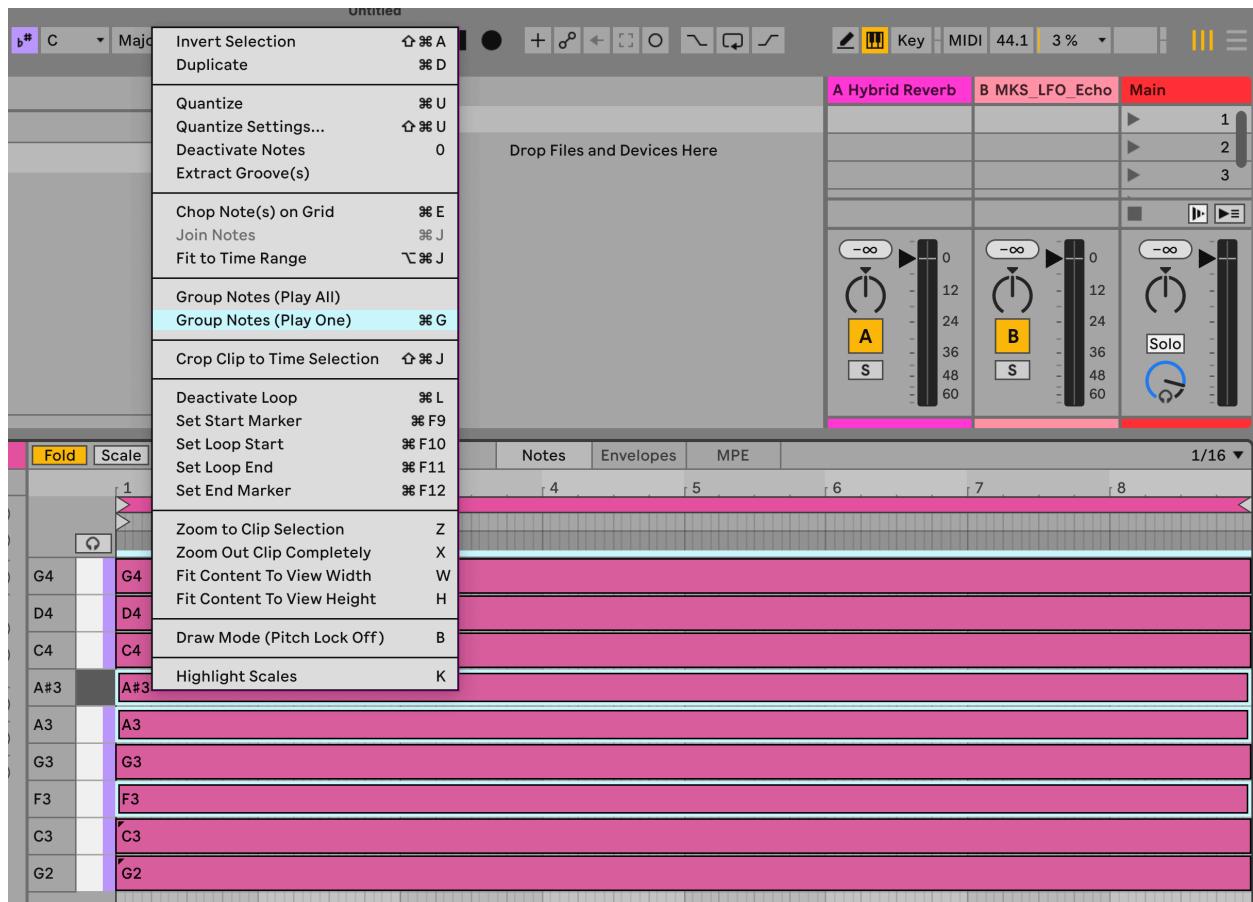


We'll look at MIDI Generation and Transformation tools a little more in [Tools - Live 12 MIDI Generation and Transformation](#).

## 24.3 Note probability

In [Process - Note-wise probability](#), we looked at Notewise probability - where we can set the probability for each note in a sequence, so we can generate ever changing patterns even from fairly short sequences of MIDI notes. Live 12 has another fantastic feature where we can **group** notes together and either select to play ***all*** notes in the group and adjust the probability of that group of notes, or group notes and select to play ***only one*** of the notes in that group. What this means for the Lazy Producer is that you can specify chords that when triggered play all of the notes (probability acts on the ***chord***) or you can nominate a group of notes where the computer picks which note in the group will play, but you can be sure that exactly one of those notes ***will*** be played.

## 24 Tools - Live 12 Overview: MIDI and compositional tools



Let's give a concrete example. Imagine you want to set up an arpeggiated pattern, but you always want the pattern to play four notes. You always want the root note of chord and the fifth of the scale to play, but you want the other two notes in the pattern to vary each time the chord is played. Well you would group the root and fifth and select "Play All", while you could group the other notes in the chord and select "Play One". This will maintain a steady pattern and predictable notes (root and fifth of the scale) while other notes provide some randomness and interest. Note that these actions on the grouped notes (Play One, Play All) also follow whatever is set for the probability setting for those notes / groups. So you can have a group of notes with "Play One" but that one note will only happen 50% of the time. This opens up enormous flexibility in generating note patterns or sequences with a LOT of variation.

# 25 Tools - Live 12 Sequencers

Live Suite has had Max for Live sequencer devices for some time, but with Live 12.1 some of those have now been included in a new Sequencers pack. The sequencers in this pack allow you to specify a pattern of notes, octaves, velocities, probabilities, and many other attributes and sequence these in ways that are much more like modular system sequencers, where pattern lengths may differ from number of beats in a bar, or where you can randomise aspects on the fly.

## 💡 Key idea

Why put sequencers in a DAW (which is essentially a sequencer)? Well, because then you can specify notes you want to play, but have these play in a nonlinear way, or not follow the overall time signature of your Live set. It opens up possibilities of playing chosen notes but vary the order these are played resulting in patterns that vary but in very interesting, evolving and generative ways.

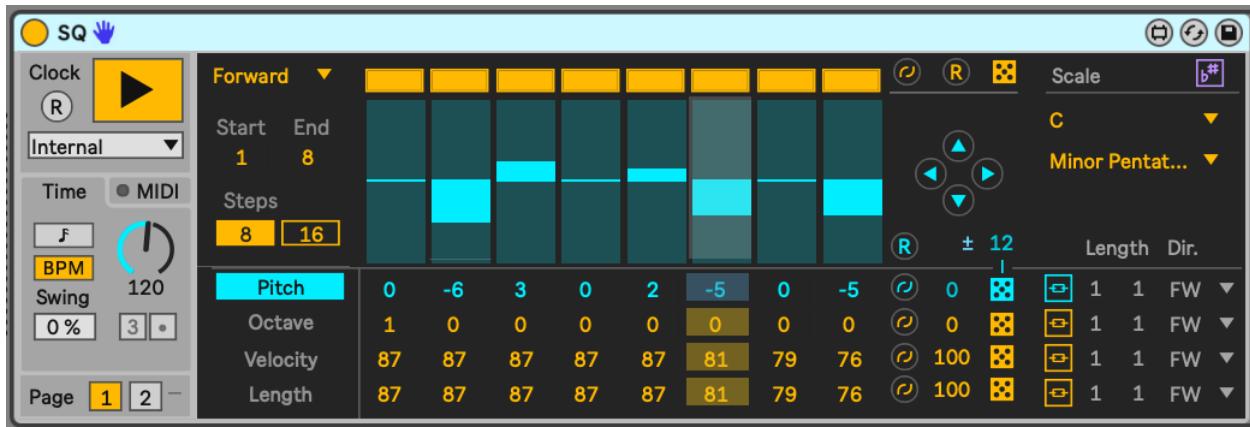
One limitation of MIDI is that it defines both the note on and note off information as well as the pitch. In the modular synthesizer world, pitch sequence and gate (whether the note occurs or not) sequence can be completely independent, and most interestingly can run at different rates or have different lengths to create permutations of patterns. The sequencers in this pack offer a little bit of the modular experience within Live - where pitch, velocity, octave and other attributes can have different sequence lengths. In order to get the “gate” sequence with MIDI you can set the velocity to zero to specify that a given note should not sound.

## 25.1 SQ

The SQ sequencer offers a HUGE range of tweakability to specify note pitch, octave, velocity, length, time-shift, ratchet and condition and have the sequence of values for each of these run alongside the pitch sequence or operate with a different sequence length to each other. So for example you could specify a pitch sequence, then have the note velocities change and rotate with a shorter sequence length to add some variety. Click on the “link” icons to the right of the note attributes (e.g. Velocity) to unlink the sequence for that attribute from the pitch sequence length. Clicking on the “dice” icon will randomise values. The “Length” attribute defines the length of the sequence, while the “Dir” setting defines the sequence play order - the sequences also don’t need to run forward, but can run in reverse, ping-pong back and forth (running forward then backward then forward), in a “snake” pattern or randomly.

You can have the sequence be scale aware so that notes respect the Global Scale setting. You can change the starting note of the sequence by playing a different MIDI note. You can set the clock for the sequence to run at a different rate to the Global BPM setting in the Live set, or even set the clock from an external device. MIDI pitch information can be entered through the Pitch lane, or can be entered via a Step Sequenceer within the MIDI tab.

Note (as above in the Key Idea) that setting velocity to zero will stop the MIDI note from sounding, so if you would like to have a “gate” sequence, then you can specify notes which notes not to trigger.



This is a really powerful sequencer then, that can provide endless variation and interest from fairly simple patterns.

## 25.2 Rhythmic Steps

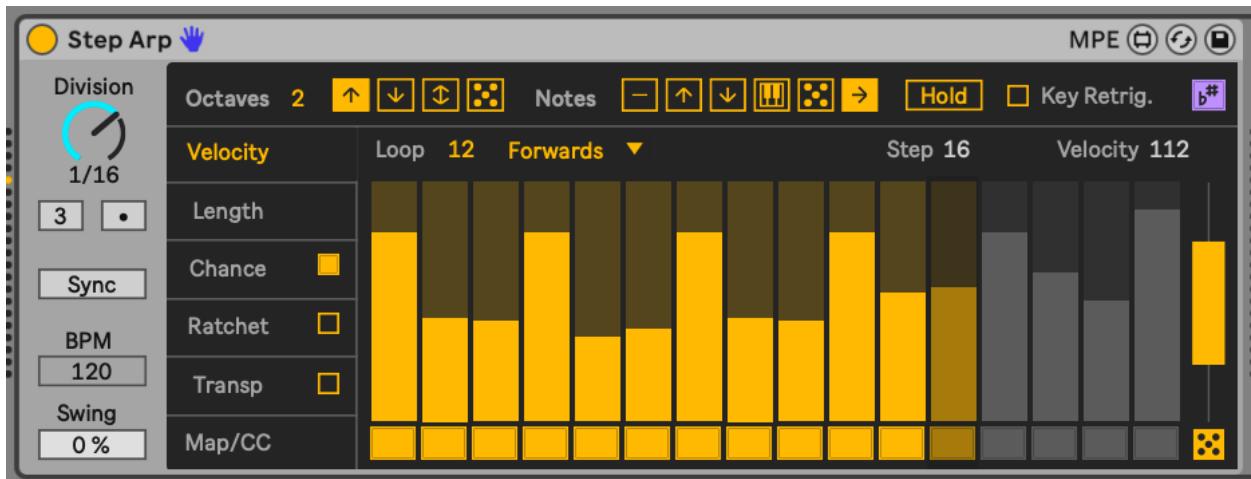
Rhythmic Steps is quite similar in some ways to the SQ sequencer, in that we can define the sequence of certain attributes - chance, velocity, MPE slide, and roll - and vary the sequence length of each of these if we wish. Rhythmic Steps is set up to work with Drum Racks (you can see which note is going to be played from the Drum Rack in the left hand side graphic). The “Alt” attribute is interesting as it allows you to specify the probability of playing an alternate sound (in blue in the Drum Rack graphic), so you could mix closed hi-hat (second from top row) with an open hi-hat sound.



You can also reset the pattern (using the [R] or [Auto] buttons in the image above) so that the patterns will start back at step 1 again. The [Auto] setting allows you to specify a number of bars after which the reset will happen if you don't want to manually trigger this. The icon to the left of the [R] reset button shows how the Rhythm Sequencer can be used directly from Ableton's Push controller.

## 25.3 Step Arp

Much like the above sequencers, Step Arp allows you to tweak and sequence the attributes of an arpeggiator and have the sequence of those attributes run in parallel with the pitch sequence which is governed through the Octave and Notes button settings or unlinked to sequence the attributes with different lengths.



The bar to the right of the sequence allows you to specify a range in which the values will be randomised, so here this would affect the range of velocities that would be randomised.

# 26 Tools - Live 12 Modulation

Live 12 has now introduced the concept of parameter **modulation** through devices like the LFO effect, and others. Modulation (instead of **control**) opens up new possibilities for the Lazy Producer.

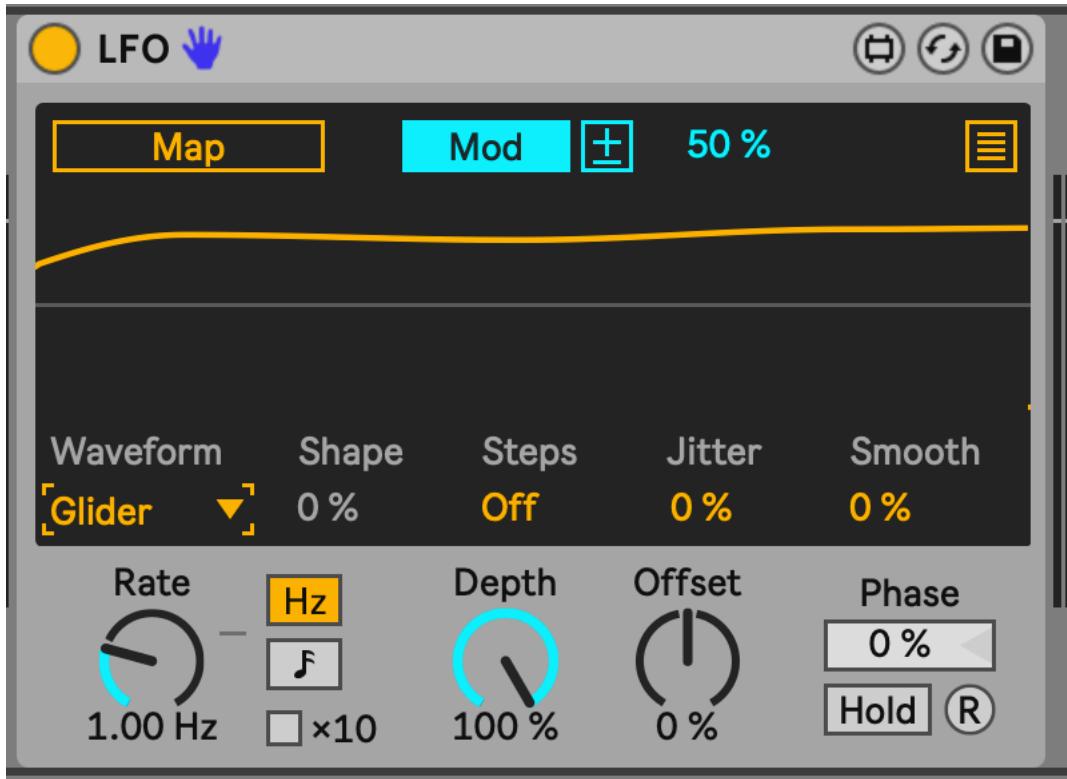
## 💡 Key idea

Previous versions of Live Suite allowed you to change values of parameters through the (Max for Live) LFO device. In Live 12 this feature has been extended by allowing you to **modulate** a parameter by a given amount through the LFO. What this means in practice is that you can continue to change the value of the whatever parameter you are modulating, but the LFO will apply additional changes to that parameter relative to its current position. Previously, when the LFO controlled the parameter, you gave up complete control over that parameter to the LFO device, but now you can continue to make changes, while the LFO adds a relative change around the chosen value.

## 26.1 Remote control or modulate?

Next to the “Map” button in the new Live 12 LFO is the choice to Remote control a parameter which means that the LFO will take control of the parameter and change it according to the settings in the LFO device; or Modulate the parameter leaving you to change the parameter value but with the LFO changing the value from where it is currently set. The modulation can be “unipolar” meaning that the values will go up *or* down (but not both up *and* down) or “bipolar” meaning that the modulation will go up *and* down by the set amount. To set the LFO to modulate downwards, click on the “+” or “±” button next to “Mod” and then click and drag the % value box until it shows negative numbers.

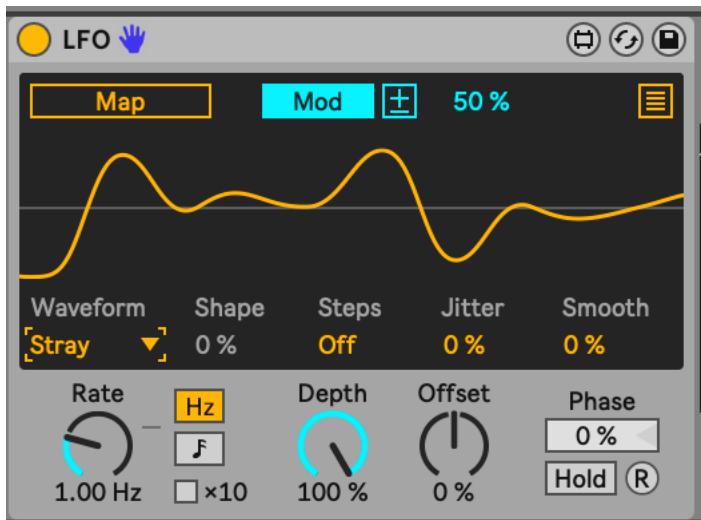
## 26.2 Live 12 - LFO shapes and modulation



In the screenshot above, the LFO is changing the value of the mapped parameter by 50% relative to its current value.

Also, note the new LFO “Glider” type in Live 12. This gives a really useful random, but smooth change in values. Previously to get a smooth but random LFO, you had to chain LFOs together and cross-modulate the amount or rates (as discussed in [Tools - Modulators](#). “Glider” Waveform makes that much easier.

Another LFO Waveform - “Stray” - gives similar random but smooth changes, but with a faster rate of change. Personally I favour these “random but smooth” changing LFOs, as the previous “sample and hold” random LFO was a bit too “steppy” for my liking.



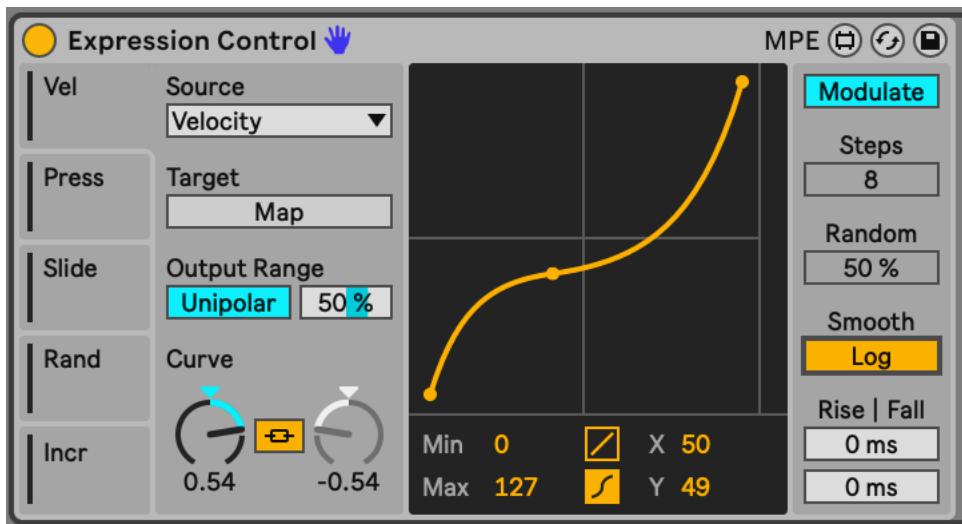
There are a couple of additional options in the Live 12 LFO - Shape and Steps. The Shape amount allows you to alter the shape of the LFO waveform, skewing (in Triangle waveform), flattening the peaks (in Sine waveform) and other ways of altering the overall shape of the waveform. This could be useful when you want a periodic shape, but don't want the modulation to stick to traditional waveform shapes. Steps essentially "bit reduces" the waveform, and introduces more stepped forms. This might be useful in conjunction with the beat synced mode where changes in modulation amounts could occur on 8th or 16th note divisions.



You can apply LFO modulation directly to parameters of an instrument, effect etc. but one useful trick is to then group the device(s) and map a Macro knob to the "Depth" of any LFO or modulation devices. This will allow you to dial in more or less modulation from only one knob, rather than having to individually change a number of settings. The combination of having modulation devices change parameter values relative to the current setting, and the ability to dial in more or less of this modulation allows the Lazy Producer (and Lazy Performer) to create sounds that are static, but also sounds that are constantly evolving. This is great for keeping the listener engaged.

## 26.3 Live 12 - MIDI Expression Control

The UI for Expression control has changed somewhat in Live 12. The functionality is largely the same, except that now (as with the LFO) we can **modulate** a mapped parameter and not just remote control it. This means that we can specify an amount of modulation on the mapped parameter for the expression to control, but the parameter itself can be tweaked by hand to set its level.



## 26.4 Live 12 - Shaper

The Shaper modulation device remains much the same in Live 12 as in Live 11, and features the modulation capabilities, similar to LFO and MIDI Expression Control above. However in Live 12 there are some additional capabilities for specifying whether the Shaper modulation loops, is a One Shot or is triggered Manually.



# 27 Tools - Live 12 MIDI Generation and Transformation

During the process of writing this book, Ableton released Live 12. With Live 12 there are a whole host of improvements and additions which make the life of someone making generative music a whole lot easier. In this chapter we'll look at the new compositional and MIDI tools. In the next chapter we'll look at new sequencer tools. In subsequent chapters we'll look at updates to modulation, and some specific Transformer and Generator tools.

## Key idea

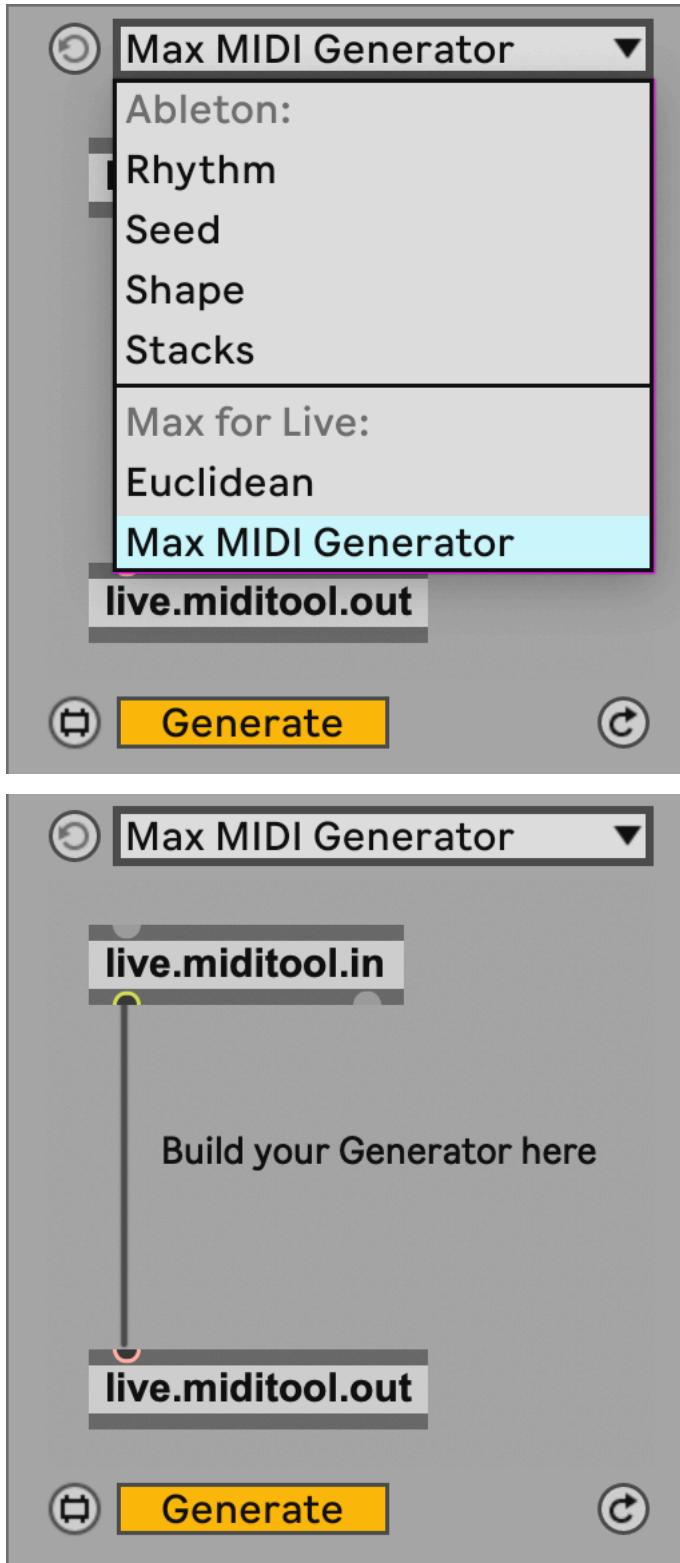
Live 12 contains a number of tools and new features which enhance how we can make generative music using Ableton Live. Some of these are enhancements to existing functionality, but there are many places where there are brand new features which could be used to create generative music, or provide idea generators for composition.

In particular note that you can Generate, commit to a pattern, then Transform the MIDI multiple times. This allows you to “stack” Transformations - for example, generating a pattern, then using Transform to change pitch, commit changes, then use a different Transform tool to change velocity, then use Transform again to change other elements, such as note division or the placement of notes in the clip.

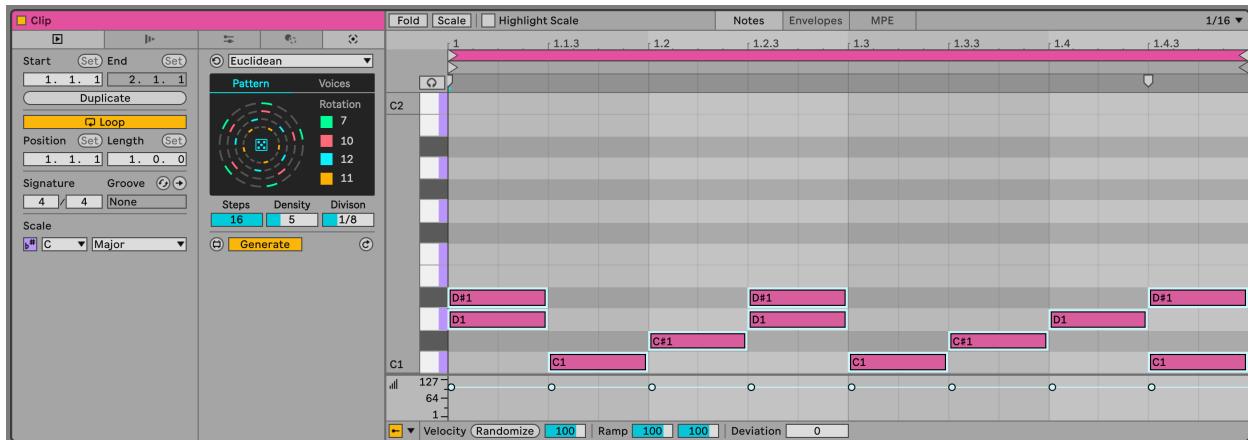
Exploration of the combination of Generation and Transformation is encouraged to get the most out of these tools. You shouldn't view them as single tools that have to do everything in one go. Often the layering of Transformations will give the most interesting results.

## 27.1 MIDI Generation

Live 12 now comes with two new features built into MIDI clip creation: MIDI Generation and MIDI Transformation. MIDI Generation does what the name suggests - provides a range of different methods for generating new MIDI parts: rhythm - which creates sequences of MIDI notes by specifying the density of notes within a certain number of steps and then allowing you to choose permutations of these, stacks - which defines chords, seeds - which generates one or more “voices” of completely random notes (but these could conform to the clip scale setting) within the clip, shape - which generates notes by drawing in a pattern or shape, and Euclidean generators. And because what sits behind these generators is the power of Max for Live, the possibilities open up for a huge range of innovative MIDI generation.

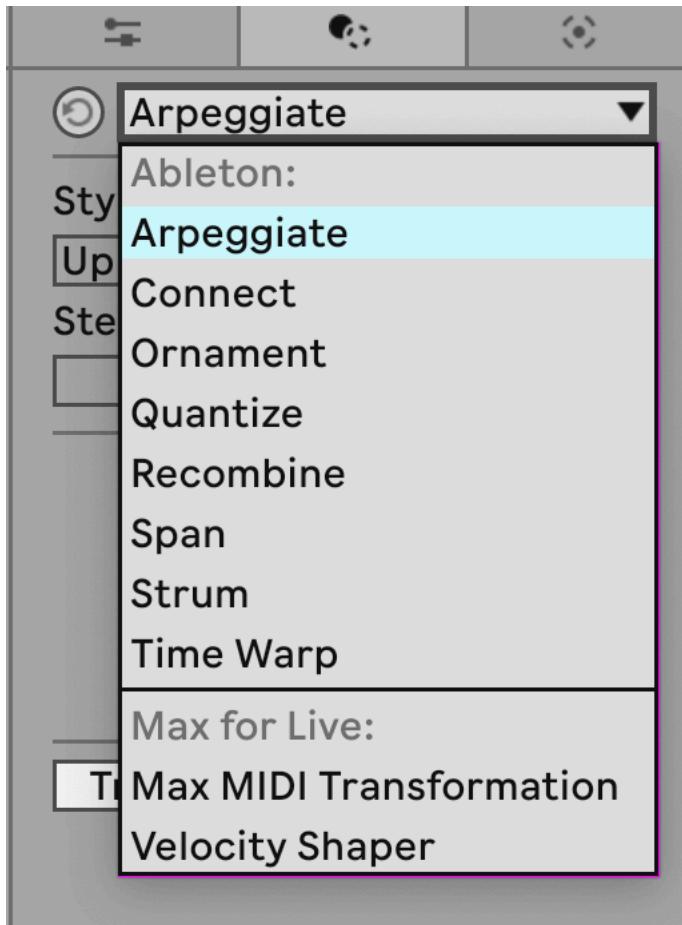


In Chapter 13 we looked at MIDI Generator devices including Max for Live devices. But having some similar functionality here “baked into” the Live MIDI clip tools puts these tools at the fingertips of anyone with Live 12 Suite.



## 27.2 MIDI Transformation

The MIDI Transformation tools includes tools to arpeggiate the notes within the clip, connect - connecting existing notes in the clip with passing tones, ornament - adding additional short notes before the main note, like a flam in drum terms, or mordent in classical notation, quantize - here applying quantisation non-destructively, i.e. you can see the effect of the amount of quantisation on the MIDI notes, recombine - which lets you select notes and then permute these (rotate the order, mirror, shuffle), span - which allows you to change the note lengths by various means, strum - which applies a rhythmic strum pattern to chords, and time warp - which can bend the timing within a bar, so although the bpm clock remains constant, MIDI notes in a clip can trigger with a more fluid timing.



Some of these Transformations will give really interesting results on pitched instruments, while others might be more suited to unpitched or rhythmic parts. It's worth exploring to see how they apply in different contexts.

Because the tools apply to creating and transforming MIDI in clips, they are really useful in the initial stage of production and composition where you want help to create something new. They can also help you create variations of existing clips, taking one good idea and evolving it into something completely different.

## 27.3 Limitations

The limitation of these Generator and Transformation devices as they stand is that their processes apply to a MIDI clip and cannot be triggered remotely in performance or through clip automation. So while it's possible to randomly generate MIDI sequences, melodies, etc. the typical use for this is at the time of MIDI clip generation. It *is* possible to regenerate for a clip, but it would be tricky to do this during a live performance. So these tools might be thought of as tools to get started and generate ideas, rather than tools for generative performance.

## 27.4 Extensions

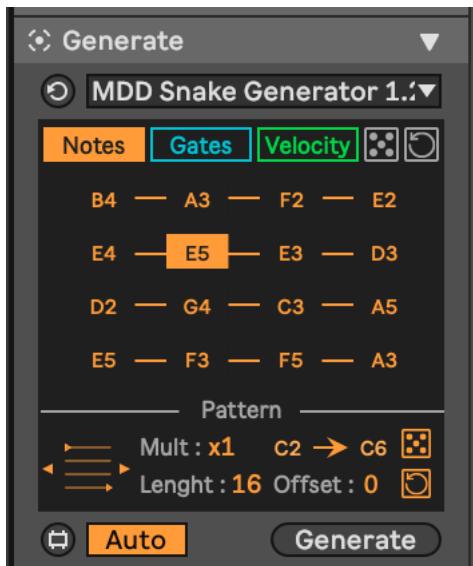
Since these Generator and Transformer devices support Max for Live devices, it has opened up the doors for many developers to create really interesting new Generator and Transformer devices. One developer of note is Philip Meyer - <https://meyer-devices.com> - who has a collection of devices that enhance, evolve and augment the built-in Ableton Generator and Transformer devices.

The beauty of Max for Live with Ableton Live Suite is the ability to create these kind of extensions and tools that integrate so well with Live's own internal devices.

### 27.4.1 Generator - MDD Snake

In [Tools - MIDI Generators](#) we mentioned the MDD Snake Max for Live MIDI generator (<https://maxforlive.com/library/device/10422/mdd-snake-midi-generator>) . This is a great sequencer modelled after the Make Noise René modular sequencer. The device gives you a four by four grid of notes, gates and velocities and it cycles through each of these in a given pattern. The default plays each row before moving on to the next row. But you can also choose to play through the notes column by column, in diagonals, and a variety of other patterns. Each pattern can play through forward, backward, cycle forward and back etc.

The beauty of this little generator is that it decouples note pitch from gate (whether the note plays or not). So you can generate patterns of notes / pitches separately from the pattern of whether the note plays which avoids the machine gun pattern of constant 8th or 16th notes.



There's an interview with Maxime Dangles, creator of the MDD Snake device here: <https://www.ableton.com/dangles-unlikely-connections/>.

### 27.4.2 Generator - Turing Machine

We've also discussed the Turing Machine in [Tools - MIDI Generators](#). Turing Machines create random sequences of notes and this MIDI Generator Max for Live device by Philip Meyer is essentially a little Turing Machine that will create a clip full of notes for you. Because the Turing Machine produces random note patterns, it's worth experimenting and generating a few patterns until you find one that works in context. Philip Meyer explains the device in more detail in this video: [https://youtu.be/\\_LpA66peATg?si=-KgyOTrVt\\_-sbTgv6&t=2191](https://youtu.be/_LpA66peATg?si=-KgyOTrVt_-sbTgv6&t=2191).



## 27.5 Philip Meyer MIDI Tools

Philip Meyer has created a number of Generator and Transformer MIDI tools which are available via his website: <https://meyer-devices.com>. In the video linked above, Philip demonstrates each of these and it's worth reviewing the content of that video to both understand how the devices work, but also to see how he Generates, then Transforms clips.

### 27.5.1 Transformer - Develop

The Develop tool, which takes a phrase and then over a set number of generations (copies) it can include more notes with each generation (or Degrade to eliminate notes) through muting notes in the pattern. In the screenshot below notice how in the first repeat of the pattern, about 50% of the notes are muted, and then as the pattern repeats, more notes are unmuted until the final repeat has all notes playing.

The Deterministic mode mutes notes in the pattern while a Probabilistic mode reduces probability over generations. In Probabilistic mode you can also set a percentage (euclidean

## 27 Tools - Live 12 MIDI Generation and Transformation

pattern) to specify which notes in the pattern have lower probability. This is a really creative tool to introduce variation in note patterns so that they evolve rather than change suddenly.

Again, Philip presents the tool in this video: [https://youtu.be/\\_LpA66peATg?si=pOATIb8VZp7j3eub&t=](https://youtu.be/_LpA66peATg?si=pOATIb8VZp7j3eub&t=)

