# SuperCollider Addons i'd recommend

## See Recommded Addons.scd for code examples

TODO: Add link to harmonic series patterns and Pkr when done

Here is a list of Extensions and Quarks that are crucial to my live performances. If you want to be able to use all of the resources in this repo, you should install them.

### Extensions

Extensions have to be inserted into SuperCollider manually. See this document for more information. Note sc3-plugins have to be compiled on Linux. See the sc3-plugins readme on GitHub for more information.

### sc3-Plugins

'This repository contains the community collection of unit generator plugins for SuperCollider. An installation extends the functionality of SuperCollider by additional UGens that run on scsynth, the SuperCollider audio synthesis server.'

sc3-plugins is a mixed bag of tools, and contains a lot of things I don't use, but it's pretty essential for expanding the functionality of SuperCollider. A good few of the sc3-plugins are fairly scantily-documented, and fall into the 'sounds cool, but no idea what it does' category.

Particular tools from sc3-plugins I use regularly:

- Concat and Concat2 Tools for concatenative synthesis. Particularly useful when dealing with speech and sampling - i've used them to 'reconstruct' speech using existing samples.
- Decimator and SmoothDecimator Bitcrushing effect Ugens
- SawDPW (and PulseDPW) Alternatives to SuperCollider's native Saw and Pulse Ugens, which alias much less, use less CPU and sound an awful lot better especially during additive synthesis. Can also get really wild at unusal frequencies.
- DFM1 A really fantastic sounding digitally-modelled analog filter. Really great both as a scuzzy-sounding filter when pushed into self oscillation to make drones. Sounds good both in moderation and absurdity.
- CrossoverDistortion A savage distortion
- WaveLoss An effect for dropping sections of waveforms in either a deterministic or random fashion. Produces a 'degredation' effect from slight droupouts all the way to isolated spluttering.

Note for installation: My /path/to/sc3source is usually /usr/local/include/SuperCollider

### BenoitLib

A set of SuperCollider extensions used by Benoît and the Mandelbrots.

The main tool I install this for is Pkr, a pattern proxy for synchronising control rate Ugens inside of patterns, which is a technique I will be covering in this repo. It's a small part of the extension but is totally invaluable for my performances.

There's also some super useful stuff in BenoitLib for collaborative performance which I have used before in a performance with Shelly Knotts, including MandelHub and MandelClock

### Quarks

Quarks can be installed from within supercollider, either by installing them manually (`Quarks.install('BatLib')` for example), or using `Quarks.gui` to bring up a gui install them there.

### Bjorklund Quark

The Bjorklund quark implements Euclidean Rhythms, a concept outlined in this paper, involving taking a number of onsets and a number of possible steps, and spaces out the onsets as equally as possible in the given number of steps. A verbal explanation of this doesn't really do it any justice, so i'd encourage you to use this cool web app which visually and aurally explains what these rhythms are. I've found Euclidean rhythms a great way to program rhythm that is dynamic and interesting, but also sits well within a set of metric dance music. The class I use from this quark is Pbjorklund2, which gives an array of durations for euclidean rhythms.

### BatLib Quark

BatLib contains StageLimiter, a class that puts a basic limiter across all sounds in the SuperCollider server. StageLimiter doesn't really have any effect on the sound the server makes unless you exceed an amplitude of +/- 1 (the top and bottom of the default SuperCollider scope), and when you do push harder than that, you can use it creatively to get 'sidechaining' type effects. I'd recommend always running StageLimiter unless you have a specific reason not to anyway, as an amplitude value that is accidentally out by a factor of ten can be *really* painful.

### ddwSnippets Quark

ddwSnippets is a ‘Rudimentary snippets facility for ScIDE, implemented in sclang’. I’ve found snippets are very useful for any piece of text that will be typed multiple times during a performance, or to lay the groundwork for ‘basic’ musical patterns without having to write them from scratch (see my comments in 0-2 about SuperCollider’s verbosity). I use ddwSnippets to realise musical ideas more quickly when performing, especially using Ugens or patterns that have a lot of arguments, without having to copy-paste from a ‘template’ file containing the snippets.