BRIEF EXPLANATION OF THE PATCH ELEMENTS

SOURCE INPUT AND RESULT OUTPUT

input\_sequencer

This is the source sequence player for the patch. Using the Max seq object, this input sequence player approach does not play or record all midi metadata messages (timesig change, tempo change, key change, etc.

output\_sequencer

This is the final output recorder sequencer for the patch. Using the Max seq object, this output sequence recorder approach does not record or play all midi metadata messages (timesig change, tempo change, key change, etc.

monitor\_output

This allows for monitoring the input and process(s) without recording to the output sequencer.

PATCH ROUTING

routing\_pb

Menu-driven module for selecting processing series. The output of any process can be the input of any other process.

start\_process

Triggers “on” and “off” states in the selected processing modules in the processing series.

GLOBAL UTILITIES

all\_notes\_off\_module

This turns off all notes on all 16 MIDI channels. This takes approximately 2 seconds to complete.

sustain\_pedals\_off

This turns on/off all controller 64 sustain pedals on all 16 MIDI channels. This takes approximately 2 seconds to complete.

global\_chord\_select

For those processes that use chord selection, this sets the chord globally to multiple modules simultaneously.

audio\_playback\_module

This plays a pre-recorded audio file. This is only for monitoring. The audio is not used in any processing.

UTILITIES

multi\_layer\_merging\_all\_channels (first created pre 2010)

This process enables the user to merge two or three sequences with three degrees of priority over the others for overlay. When a note in a track (channel) exceeds the threshold time the attention of the merger is directed to the lower priority sequences. The result is long notes and rests in the higher-priority sequence being replaced by events from the lower-priority sequences.

multi\_layer\_merging\_single\_channel (first created pre 2010)

This process enables the user to merge a single channel (track) from two or three sequences with three degrees of priority over the others for overlay. When a note in a track (channel) exceeds the threshold time the "attention" of the merger is directed to the lower priority sequences. The result is long notes and rests in the higher-priority sequence being replaced by events from the lower-priority sequences.

sequence\_concatenation (first created pre 2010)

This process uses an elapsed time clock to schedule controlling events to interleave a collection of prepared MIDI sequences. The specific programming in this version is for the construction of the "J" ending section of my composition Triptych, II. Innocent Proteins.

sequence\_telescoping (first created pre 2020)

This process enables the user to telescope up to three MIDI sequences or three instances of the same sequence to compress the timing with forward jumps in the original sequences or expand by overlapping segment endings and beginnings. The overlapping seq objects can play the same file and compress and expand the combination by identifying segment lengths and overlapping the next segment start with the currently playing segment (or space them out with rest for expansion).

remove\_hanging\_notes

This process removes notes from sequence that might be hanging up by flushing all notes before each new note in each track.

velocity\_to\_PC\_contour\_dynamic\_range (first created pre 2010)

This process converts note event velocities to a PC curve to graphically indicate dynamics and offers hi-low analysis of velocity range.

sequence\_to\_difference\_tables (first created pre 2010)

This process analyzes difference between successive integer events from input stream converting incoming sequence notes to tables for pitch difference (interval), velocity difference, and duration difference for use in other processes and analyses.

sketch\_rough\_track\_filler (first created pre 2010)

Play seq single-channel (ch 1) MIDI file and have the notes that fit ranges of instruments sent to those channels on the output multi-channel sequence. \* This is NOT orchestration! Impossible/impractical music can/will result \* Currently supports output to 32 MIDI channels using two 16-channel sequences. output to 33+ MIDI channels are for additional orchestral instruments.

DEVELOPMENTAL PROCESSES

alter\_linear\_intervals (first created pre-2010)

This process allows for linear interval redefinition by testing intervals between successive note events and modifying them to conform to defined limits. This process will modify lines with extreme register leaps to more limited contours.

alter\_velocity (first created pre-2010)

This process allows for altering MIDI note velocity with a multiplication value and thereby creating rests in a sequence when velocity exceeds the set limit (if the 'reject past limit' is set for that channel).

alter\_X-Y\_intervals (first created pre-2010)

This process allows for between-channels horizontal and vertical interval redefinition (pitch changes to match other channels) by testing simultaneously sounding pitch intervals between channels and modifying them to conform to funbuf interval sets. Intervals are treated symmetrically with unison as axis.

branching\_sequence\_time\_scaler (first created pre-2010)

This process allows for layering multiple iterations of the same sequence with differing tempi (time scales) by entering scale of tempo for each overlaid sequence (.5 = twice as slow).

catch\_first\_notes (first created pre-2010)

This process filters a midi file and passes only the first note of figures that are separated by a set threshold of time (rest or sustain).

catch\_long\_notes (first created pre-2010)

This process filters out fast notes and passes only long notes to the output. All notes will be delayed by the duration of the tested long note threshold.

center\_finder (first created pre-2010)

This process finds the center of an input chord or sequence of notes. The center is reported periodically as a midi note number that is not necessarily a pitch that has been present in the analyzed collection (abstract center of the "pitch space" of the events during the analysis period. This can be used to transpose chords or sequences to share common center axis of symmetry.

chord\_builder (first created pre-2010)

This process allows an input note to trigger one of multiple chosen chords that builds the chord so that the lowest chord tone is at the input pitch.

chord\_thinning (first created 2011-2012)

This process takes input block chords and removes notes (thins) the number of notes in the chords based on specified conditions. The rejected notes can be placed into alternative (offset) channels and delayed to be played in the first empty space in the track following their thinning. This process helps to simplify the track for performance using context and history to determine what to remove.

chord\_tremolos (first created pre-2010)

This process allows for the creation of tremolos between 2-4 chords

chordal\_sequences (first created pre-2010)

This process sets up a series of chords and triggers them from midi input source.

chords\_from\_context (first created pre-2010)

This process interjects chords built on n-number of notes drawn from the immediate linear context of a sub-sequence (context sequence 1).

combine\_multiple\_sequences (first created pre 2010)

This process merges multiple sequences into one common sequence with switching and joining functions.

composite\_switch (first created 2011-2012)

This process enables the user to create composite 2-track instruments (arco-pizz, open-muted, etc.) where the switches between the tracks are prioritized and also have speed limits (to accommodate time needed for arco-pizz change or open-muted change, etc.). When a pause in a channel exceeds the threshold time the "attention" of the merger is directed to the next change in channel. This will leave 'holes' in one channel when the paired channel is active. This works with 16-channel sequences where each pair of channels are switched independently of other pairs.

condense (first created pre-2010)

For each channel of a MIDI sequence, n input sequence begins with upwardly-transposed and duration-shortened playback gradually working back to the original source sequence, eventually playing the sequence intact. This process is timed independently for each track in the source sequence.

convert\_keyswitches (first created 2011-2012)

This process takes a sequence and converts track controller data (from Finale, etc.) to Garritan KeySwitch note data. The program can also convert controller messages into note messages to handle Finale marking playback data.

crossfade\_sequences (first created 2011-2012)

This process enables switching between five multi-channel sequences with varying staggered switch timing for each track as well as a few other modifications for each sequence.

data\_stepDance (first created 2011-2012)

Uses a step-register process to move through event data based on control data from pitch interval and duration ratio analysis files.

define\_perforator\_input

When using the "perforator" process, when, in that process, "define perforator" is enabled, note events sent to define\_perforator\_input will set the process definitions for note events.

define\_sieve\_input

When using the "sonority\_sieve" process, within that process there is a subpatch "define sieve." In "define sieve" note events sent to define\_sieve\_input will set the notes of a newly-defined sieve set.

downbeat\_split\_accents (first created 2009-2010)

This process allows for splitting the notes from a sequence that fall on or just after the first beat of each measure. The split uses flexible channel shifting for downbeat accented notes. The split notes pass through a time window that can be of varying length and also can be offset to let through notes that fall on other parts of the measure than the downbeat.

edge\_of\_disorder (first created 2011-2012)

Controls the application of random mutation modification to elements of incoming sequence note events using probabilities for randomness modification of pitch, velocity, channel, and event start time for each channel

ensemble\_distributer (first created pre-2010)

This process distributes incoming events into range-determined midi event streams by analyzing for range and then evenly distributing them among four range-determined sets of four streams (total of 16 streams created).

evaporate (first created pre-2010)

For each channel of a MIDI sequence, playback is interrupted and short note events are transposed upward while getting shorter and quieter. This process is timed independently for each track in the source sequence.

fill\_tables (first created pre-2010)

This utility takes an input midi sequence and fills data tables with pitches of chords for user-selected tables 1-10, durations for tables 1-10, velocities for tables 1-10, event channels for tables 1-10, differences between sequential pitches, velocities, and durations.

foreground\_matching\_context (first created pre-2010)

This process adjusts the pitches of a 16-channel foreground sequence to match a background harmonic field context of a sub-sequence context sequence.

heterophony (first created pre-2010)

This process reads from tables for pitch, duration and velocity (created in the "table based sequencer" and plays the line heterophonically with 16 channels.

hocket (first created 2011-2012)

This process divides a midi stream up between n-number of channels one event at a time.

interval\_doubling (first created pre-2010)

This process adds intervallic doublings to the linear input sequence of pitches.

limit\_linear\_intervals (first created 2008-2009)

This process allows for linear interval modification by testing intervals between successive note events and modifying them to conform to defined limits. The default limit is one octave. This process will modify lines with extreme register leaps to more limited contours.

modify\_redundant\_iterations (first created 2011-2012)

A process that, for each track, counts the number of times a specific pitch (or pitch class) has been returned to and then transposes the sequence (or track) by a pre-set interval sequence when a threshold number of returns have occurred.

move\_and\_dwell (first created 2010-2011)

This process enables the playing of a MIDI sequence with user-defined moments of progression through the sequence and transposed looping of the sequence.

multi\_bursts (first created pre-2010)

This process writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of slowing timing and lengthening durations. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" sonorities, gradually dissolving their predominance in a texture.

multi\_layer\_merging\_tutti (first created 2010-2011)

This process enables the user to merge two or three sequences with each having higher priority than the others for overlay. When a note in a track (channel) exceeds the threshold time the attention of the merger is directed to the lower priority sequences. The result is long notes and rests in the higher-priority sequence being replaced by events from the lower-priority sequences. This works with 16-channel sequences where each channel is switched independently.

multi\_motives (first created pre-2010)

This process writes incoming pitches to a table then plays a random sequence of those pitches with varying curves of slowing timing and lengthening durations. Velocity and articulation offset can also be controlled by random number generation. This can be used to "atomize" textures created from the motive, gradually dissolving their predominance in a texture.

multi\_stretch (first created pre-2010)

This process treats the playback and transposition of multiple occurrences of the same sequence in a fluid and self-similar manner.

natural\_harmonics (first created 2010-2011)

This process enables splitting to separate tracks those pitches of four strings that can be natural harmonics on a bowed string instrument.

nested\_patterns (first created pre-2010)

This process creates four-level-deep recursive patterns with transposition of each occurrence of the motive based on the pitches from incoming note events. The recursive patterns are built from interval (difference) tables for pitch, velocity, and duration.

octave\_displacement (first created pre-2010)

This process performs octave displacement by randomly adding and subtracting octaves from a note stream on a single midi channel dramatically expanding its register use.

pedal\_control (first created 2011-2012)

This process applies automated pedal events to tracks to simulate free pedaling by a performer but is based on timing between down pedals rather than a knowledge of phrasing for each track.

perforator (first created pre-2010)

This process removes notes periodically from each of 8-channel tracks and can replace them either with a defined pitch or silence (rests), or temporally truncate the track by removing notes periodically and shifting all later events to fill the gap from the removed notes.

pitch\_gravity (first created 2011-2012)

This is a process whereby a set pitch, or a pitch from a linear source stream, gradually takes predominance in all tracks of a sequence. If the imported PC centering option is selected, a track routed from the patch bay can be used to create a dynamic gravity, essentially drawing the other tracks to eventually double it at unison and octaves. The timing of the process is independent for each track in the source sequence.

pitch\_velocity\_analysis (first created 2012-2013)

This process provides a graphic histogram analysis of the note-on pitches and velocities in a sequence. There is also an option to convert velocities to quantized velocities with eight steps from 1 - 128 correlating with 8 levels of notation dynamics from ppp - fff.

pitches\_play\_on\_rhythm (first created pre-2010)

This process uses a process of multiplying prime numbers by other prime numbers to create six parallel duration scales that can be used to time the performance of a pitch sequence defined by tables of pitch midi numbers. The duration sequences are quantized using a table.

prevalence\_transposition (first created 2010-2011)

This process transposes incoming notes in a sequence to match the pitch of the prevalent pitch class from its own recent past. The track's pitch dominance is then reflected in the transpositional mutation of the track.

quantize\_rhythm (first created 2010-2011)

This process moves rhythmic events to align with a quantization grid defined by the user. This is a relatively basic approach to rhythmic quantization and cannot interpret mixtures of tuplets and duple subdivisions at the same time.

range\_manipulator (first created pre-2010)

This process takes an input sequence and converts the range under the control of width and center note. Current version supports channels 1-8.

range\_test (first created 2010-2011)

This process enables either filtering or fixing notes that are out of an instrument's range (transposing them into the range by octave transposition) for each channel of a sequence.

recursive\_series (first created pre-2010)

This process uses a table of intervals to create a series of pitches. The series reflects the interval table on two levels: the series of pitches produced by the interval table and the longer series of pitches; whose first note is the next note of the series on the lower level.

remove\_tied\_notes\_in\_chords (first created 2012-2013)

This process identifies input block chords in a sequence and removes notes that are tied over when a new note event occurs after that chord. This is used to fix situations in which a second block chord contains a new note but also has tied one or more notes over from the last chord.

repeat\_doubler (first created pre-2010)

This process senses incoming repeated notes and adds notes above or below the original as a means of brightening the spectrum and texture of passages with repeated notes.

repeats\_become\_sustain (first created pre-2010)

This process takes an input sequence and replaces repeated notes with sustained notes or removes them from the sequence replacing with silence (rests).

replace\_from\_parallel\_sequence (first created pre-2010)

For each channel of a MIDI sequence, pitches and/or velocities are replaced with currently sounding pitches/velocities from another parallel-playing sequence.

scale\_linear\_intervals (first created pre-2010)

This process allows a line to be compressed and expanded based on the %-scale setting. A setting of 1 will output an unaltered line. A setting of 0.5 will compress each intervallic change by half. A setting of 2 will expand each intervallic change by 2. Linear changes can also be inverted using the norm/invert setting.

sonority\_sieve (first created pre-2010)

This process takes an input sequence and moves pitches to match the set of pitches defined in the chosen sieve. For example, if a chromatic sequence is input and the sieve is set to “C major triad” then the output sequence will change all notes not in the C major triad to the nearest higher note of the C major triad. This allows for mutations of sequences to match either static or changing harmonic ‘fields’ for the output.

sound\_planes (first created pre-2010)

This process plays sustained tones when the user presses the "p" key to control timing and duration. Pitches are taken from a selection of six pre-loaded pitch tables (tables for chord1-chord6). Alternatively, input notes from the input sequencer routed through the menu matrix can also be used to trigger each tone.

speed\_limiter (first created pre-2010)

This process allows for linear sequences to have a maximum event/sec density by limiting how many events can pass from an input sequence per time period set by the user (in ms).

split\_accents (first created 2010-2011)

This process allows for splitting the softest and loudest notes from a sequence. Uses lower channels for non-accented and upper channels for accented notes. This process only looks at the velocity and not a comparison with average velocities in the time context. See 'pitch\_velocity\_analysis' module for that contextual approach to accent splitting.

split\_downbeats (first created 2012-2013)

This process allows for splitting the notes from a sequence that fall on or just after the first beat of each measure. The split uses flexible channel shifting for downbeat accented notes. The split notes pass through a time window that can be of varying length and also can be offset to pass notes that fall on other parts of the measure than the downbeat.

step\_dance (first created 2013-2014)

This process uses a step-register process to move through event data based on control data from pitch interval and duration ratio analysis files by controlled stepping (forward and backward) through note event data to produce mutated sequence combinations.

sustain\_becomes\_repeats (first created pre-2010)

This process takes an input sequence and outputs repeated notes where there were sustained notes in the original line. The speed/duration of the repeats can be dynamic based on selections from a duration menu of choices. Repeats can also be pitch-varied for tremolo and trill patterns.

timbre\_distributor (first created pre-2010)

Two methods for distributing sequence note events to multiple instruments.

1) This method creates cross-orchestrational distributions between eight instruments (MIDI 1-8). Instrumental selections are manipulated by a wave shape controlling the overlapping distribution of events throughout the collection of instruments.

2) This method creates cross-orchestrational distributions between sixteen or 32 instruments. Instrumental selections are manipulated by rotating tracks through a set of instrumental assignments controlled by the urn object (all options must be used before choosing an option a second time)

time\_delayed\_channel\_shuffle (first created pre-2010)

This process takes a multi-channel sequence input and delays each channel by independently set times resulting in a shuffled splaying of the original sequence.

track\_add\_CC\_expression\_shaping (first created 2024-2025)

track\_add\_CC\_expression\_shaping

This process enables independent and time-varying continuous controller (CC 1, CC7, CC8, and CC11) data merged into each track. This can add realism to sustained note events, which can otherwise be unnaturally static. Shaping is determined from menu options and the selected pattern recurs after a minimum pause duration (rest) in each track. Scaling, variation, and offsets can also be adjusted.

track\_channel\_assignments (first created 2011-2012)

This process reassigns input sequence channels to user-specified output channels for each of 16 MIDI tracks.

track\_dynamic\_shaping (first created 2011-2012)

This process enables independent and time-varying dynamic shaping by track through control of note event velocities. Shaping is determined from menu options. and the selected pattern recurs after a minimum pause duration (rest) in each track

track\_swap (first created 2011-2012)

This process enables independent and time-varying channel swapping between two channel pairs for a multi-channel sequence. Swapping can also be done manually using keyboard keys using keys 1-"=" and Q-R for swapping.

track\_transposition (first created 2011-2012)

This process enables the independent and time-varying transposition of tracks of a midi sequence.

track\_work\_sharing (first created 2015-2016)

For tracks with high density of events or complexity, this process shares the work with one other track through consideration of density in the original track and rests in the co-worker track. The input sequence is alternated between the original track and the shared track to reduce the work of the original track performer.

triggered figures (first created pre-2010)

This process plays a midi file and distributes each note to one of one to sixteen channels. These notes initiate an arpeggio event in each channel that involves arpeggiating a figure based on tables for pitches and durations.

unisons\_removal (first created 2011-2012)

This process enables track-by-track removal of doubled unison notes in each track. This is useful for transcription to notation software.

use\_guitar\_open\_strings (first created 2014-2015)

This process takes all pitch classes that match a guitar’s six open strings and transposes the pitches and octaves to use the open strings. Use for increasing the performability of guitar parts.

use\_orch\_open\_strings (first created 2014-2015)

This process takes all pitch classes from an input sequence that match the orchestral string instrument’s four open strings and transposes the pitches and octaves to use the open strings.

velocity\_quantization (first created 2013-2014)

This process enables track independent quantization of velocities and replacement of a file's velocities with those from another file. This makes auto-placing dynamics in Finale simpler.

weave\_curve (first created pre-2011-2012)

This process takes a web of sequence tracks and their twelve transpositions that all begin together and play in sync. It then allows for using a mouse in a 2D grid to select from which sequence to draw events with curving mouse movements between and around tracks like a walk in a 3D weave of tracks.