

A

- bach.abs**
Quick absolute value of IIIIs
 Takes the absolute value of an incoming IIII.
- bach.adj**
Compute the adjugate of a matrix
 Computes the adjugate (or the classical adjoint) of an incoming matrix
- bach.approx**
Snap pitches or MIDIcents to microtonal grid
 Approximates the incoming IIII containing pitches or MIDIcents with the nearest pitches or MIDIcents in a specific microtonal grid.
- bach.args**
Manage subpatcher arguments
 Parses and retrieves the subpatcher's „normal“ and attribute-style arguments. The basic behavior of bach.args is similar to patcherargs 's
- bach.arithmser**
Arithmetic series
 Compute arithmetic series
- bach.autoscale**
Linearly rescale an IIII automatically
 Linearly rescales all the numbers in an IIII so that the minimum

B

- bach.bachtree2omtree**
Convert a bach rhythmic tree into an OpenMusic rhythmic tree
 Converts the measureinfo and the duration rhythmic tree (in bach form, as outputted by bach.score)
- bach.beatbox**
Box voice-wise parameters into measure-wise information
 Adds a parenthesis level representing measures in a plain list of parameters.
- bach.beatunbox**
Flatten measure-wise parameters into plain voice-wise information
 Removes the measure parenthesis levels in the bach.score parameters syntax.
- bach.belong**
Tell if some elements belong to a IIII
 Tells if one or more elements belong to a given IIII.
- bach.bin**
Compute binomial coefficient
 Computes the binomial coefficient $\binom{N}{K} = N! / (K! * (N-K)!)$
- bach.bitdecode**
Decode a bit field
 Tells which bits are on in a given integer.

C

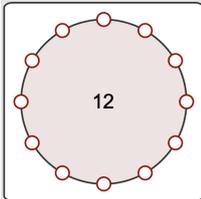
- bach.cartesianprod**
Cartesian product
 bach.cartesianprod returns the cartesian product of two or more IIIIs.
- bach.cartopol**
Cartesian to polar coordinate conversion for IIIIs
 Converts two IIIIs consisting of real and imaginary values into
- bach.change**
Filter out repetitions of an IIII
 The purpose of bach.change is the to act like a change object for IIIIs,

`bach.chordrev`**Reverse the non-rest elements of an llll**

Accepts an llll to reverse, an llll of reference durations,

`bach.chordrot`**Rotate the non-rest elements of an llll**

Accepts an llll to rotate, an llll of reference durations,

`bach.circle`**Clock diagram**

The purpose of bach.circle is to act as a simple clock diagram,

`bach.classify`**Arrange elements into classes**

Arranges elements into classes, according to a custom equality test.

`bach.clip`**Limit elements of llls within a certain range**

Constraints each element of the incoming llll

`bach.collect`**Collect elements into llls**

Groups together data it receives,

`bach.comb`**Combinations of elements**

Returns the combinations of the elements of a given llll.

`bach.combinevoices`**Combine voices into a new bach.roll or bach.score**

Takes in input a list of bach.roll's or bach.score's gathered syntaxes and

`bach.compl`**Find complement of a pitch-class set**

Given a pitch-class set, bach.compl

`bach.constraints`**Solve constraints satisfaction problems**

Solves constraints satisfaction problems.

`bach.contains`**Contents of an llll**

Returns the types of the data contained in an llll, optionally within a specified depth range.

`bach.convertbase`**Perform base conversion for numbers**

Converts numbers from any base to any other base.

`bach.coprime`**Test if numbers are coprime**

Tells if the incoming number are coprime,

`bach.coprimeser`**Coprime number series**

Finds all the numbers M smaller than the absolute value of a given number N,

`bach.counter`**Multi-index counter**

The role of bach.counter is to act as a multi-indexed version of counter,

`bach.cross`**Compute the cross product of two llls**

Computes the cross product (or vector product) of two llls having 3 coordinates.

D**bach.decode****Put an llll at the front of the low-priority queue**

Puts the incoming llll at the front of the low-priority queue.

bach.defer**Put an llll at the front of the low-priority queue**

Puts the incoming llll at the front of the low-priority queue.

bach.deferlow**Put an llll at the tail of the low-priority queue**

Puts the incoming llll at the tail of the low-priority queue.

bach.delace**De-interleave an llll**

Redistributes elements of an llll into several output lllls, in a round-robin fashion.

bach.depth**Depth of an llll**

Returns the depth of an llll.

bach.derive**Derive functions**

Computes the derivative of a breakpoint function or of a general function defined via a lambda loop..

bach.det**Compute the determinant of a matrix**

Computes the determinant of the incoming matrix.

bach.diag**Build a diagonal matrix**

Builds a diagonal matrix having the incoming llll as diagonal.

bach.dict2llll**Convert a dictionary into an llll**

Converts a dictionary into an llll, under the convention that the first element in each level is the level key.

bach.diff**Difference of two sets**

Returns the difference of two lllls.

bach.div**Quick llll division**

Performs divisions between lllls or

bach.divisors**Find divisors of a number**

Returns all the divisors of an integer number, in ascending order.

bach.dl2curve**Convert a duration line into pitch and velocity curve~ messages**

Converts the information about the duration line of a note

bach.dl2line**Convert a duration line into pitch and velocity line messages**

Converts the information about the duration line of a note

bach.drip**Automatically step through lllls with delays**

The purpose of bach.drip is to automatically output elements of an llll,

bach.dsort**Sort llll by distances**

Sorts a list depending to the distance of its

bach.dx2x**Build llll from element differences**

Constructs an llll starting from a given number

E**bach.eig****Find eigenvalues and eigenvectors**

Finds the eigenvalues and eigenvectors of the incoming matrix, via the power iteration method.

bach.enharm**Handle enharmonicity**

Modifies the enharmonicity of the incoming pitches, either by adding diatonic steps, or via automatic respelling

bach.eq**Compare two llls for equality**

Compares two llls for equality.

bach.eval**Evaluate advanced expressions**

Evaluates expressions with variables, branching, loops and user-defined functions.

bach.expr**Evaluate mathematical expressions**

Evaluates mathematical expressions upon llls.

bach.extractbasis**Find linearly independent vectors**

Extracts a set of linearly independent vectors from an incoming list of vectors.

bach.ezmidisplay**Quick way to play MIDI**

Converts the data coming from the playout of bach.roll or

F**bach.f2mc****Frequency to midicents conversion**

Converts an lll containing frequencies (in Hertz) into an identically structured lll containing the corresponding midicents.

bach.fact**Compute factorial**

Computes the factorial of a given number N, that is

bach.factorize**Factorize a number**

Returns the factorization of an integer.

bach.fft**Fast Fourier transform for llls**

Performs the Fast Fourier transform on

bach.filter**Only let specified data types through**

Takes the incoming data and,

bach.filternull**Let through every non-null input**

Takes the incoming data and

bach.find**Find elements**

Searches an lll for elements or sequences of elements satisfying a condition.

bach.flat**Decrease the depth of an lll**

Decreases the depth of an lll by eliminating couple of parentheses.

bach.float2rat**Approximate a floating point number with a rational number**

Approximates a floating point number with a rational number, within

bach.fromc&r**Reassemble non-rest and rest elements**

Accepts an llll of element corresponding to chords,

bach.funnel**Index llll elements**

Index each llll element by prepending an increasing integer to it, and wrapping them in a pair of parentheses.

G**bach.gausselim****Perform matrix gaussian elimination**

Applies the Gaussian elimination algorithm to reduce the incoming matrix to a row echelon form.

bach.gcd**Compute greatest common divisor**

Computes the greatest common divisor of two numbers or of an llll.

bach.geomser**Geometric series**

Compute geometric series

bach.geq**Compare two lllls for greater than or equal condition**

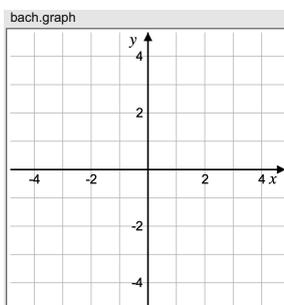
Compares two lllls for greater than or equal condition, in numerical/lexicographical order.

bach.getcol**Get a column of a matrix**

Extracts a column of a matrix in llll form.

bach.getrow**Get a row of a matrix**

Extracts a row of a matrix in llll form.

**Plot a graph**

The purpose of bach.graph is to act as a simple graph plotter: once the expression

bach.group**Group elements of an llll**

Groups the element of an llll in sublists, according to a regular size pattern.

bach.gt**Compare two lllls for greater than condition**

Compares two lllls for greater than condition, in numerical/lexicographical order.

H**bach.histo****Build histograms**

Builds an histogram starting from a list of data, returning elements along with their occurrences.

`bach.hypercomment`hypercomment ist ein comment mit [#links](#)**A comment with hyperlink references**

Works like a standard comment object, with the possibility to easily have

I`bach.idmatrix`**Build identity matrix**

Builds the identity square matrix of a given size.

`bach.idvector`**Build cartesian versors and combinations**

Builds versors having all 0's as entries, except for a 1 in a given position (or combinations of such versors).

`bach.ifunc`**Compute interval function of two pitch-class sets**

Reports the interval function of two pitch-class sets. The interval function between

`bach.im`**Find the image of a matrix**

Finds the image of an incoming matrix A, i.e. its column span.

`bach.insert`**Insert elements in an llll**

Inserts elements at specified addresses of an llll.

`bach.integrate`**Integrate functions**

Computes the integral of a breakpoint function or of a general function defined via a lambda loop.

`bach.interp`**Interpolate between lllls**

Performs interpolation between two lllls.

`bach.intersection`**Intersection of two sets**

Returns the intersection of two lllls.

`bach.inv`**Find the inverse of a matrix**

Computes the inverse of a square matrix or

`bach.is`**Type of the incoming data**

Returns the type of the incoming data.

`bach.istruct`**Get the interval structure pitch-class set**

Reports the interval structure of a pitch-class set, i.e. the list

`bach.iter`**Iterate through lllls**

bach.iter traverses lllls depth-first, outputting their elements one by one.

`bach.ivec`**Compute the interval vector of a pitch-class set**

Computes the interval vector of a pitch class-set, i.e. the vector telling

J`bach.join`**Join lllls together**

Joins lllls together, according to the positions of the inlets by which they are received.

K

<code>bach.ker</code>	Find the kernel of a matrix Finds the kernel of an incoming matrix,
<code>bach.keychain</code>	Organize IIIs by keys Classifies the elements of the incoming III, by associating them with corresponding keys, and then by outputting the III organized by keys.
<code>bach.keys</code>	Route an III according to specific keys Routes an III and its sublists according to specific keys.
<code>bach.kurtosis</code>	Find the kurtosis of the numbers in an III Finds the kurtosis (fourth standardized moment) of the

L

<code>bach.lace</code>	Interleave IIIs Builds an III by combining the elements of several IIIs in a round-robin fashion.
<code>bach.lcm</code>	Compute least common multiple Computes the least common multiple of two numbers or of an III.
<code>bach.length</code>	Length of an III Returns the length of an III.
<code>bach.leq</code> <code>bach.<=</code>	Compare two IIIs for less than or equal condition Compares two IIIs for less than or equal condition, in numerical/lexicographical order.
<code>bach.linearsolve</code>	Solve a linear system of equations Solves a linear system of equations in the matrix form $Ax = b$, where A
<code>bach.III2dict</code>	Convert an III into a dictionary Converts an III into a dictionary, under the convention that the first element in each level is the level key.
<code>bach.locate</code>	Locate elements Returns the address of the first occurrence of an element or sequence inside an III
<code>bach.lookup</code>	Return specific elements of an III (inlets reversed) Returns one or more specific elements of an III, based on their addresses.
<code>bach.lt</code>	Compare two IIIs for less than condition Compares two IIIs for less than condition, in numerical/lexicographical order.

M

<code>bach.m2jitcellblock</code>	Display the content of an III matrix in a jit.cellblock Shows the content of an III matrix in the jit.cellblock object.
<code>bach.m2matrixctrl</code>	Format an III matrix to be used with matrixctrl or matrix~ Converts an III matrix in order to pilote a matrix~ or a matrixctrl object.

<code>bach.mapchord</code>	Individually modify chords Takes the gathered syntax of a <code>bach.roll</code> or <code>bach.score</code> and operates a given transformation on each chord, defined via a lambda loop.
<code>bach.mapelem</code>	Modify llll elements Takes an llll and operates a given transformation on each element. The transformation is defined via the lambda inlet and outlet.
<code>bach.matrixctrl2m</code>	Convert a matrixctrl object into an llll matrix Converts the information contained in a
<code>bach.maximum</code>	Output maximum element-wise Compares every element of the left llll with the corresponding
<code>bach.mc2f</code>	Midicents to frequency conversion Converts an llll containing midicents into an identically structured llll containing the corresponding frequency values.
<code>bach.mc2n</code>	Midicents to note name conversion Converts an llll containing midicents into a corresponding llll containing note names.
<code>bach.mc2p</code>	MIDicents to pitch conversion Converts an llll containing cents into a corresponding llll containing pitches.
<code>bach.mc2pc</code>	Midicents to pitch-classes conversion Converts an llll containing midicents into an identically structured llll containing the corresponding pitch-classes.
<code>bach.mc2r</code>	Interval to frequency ratio conversion Converts an llll containing intervals (expressed in midicents) into an identically structured llll
<code>bach.mcapprox</code>	Snap midicents to microtonal grid Approximates the incoming midicents llll with the nearest pitches in the full chromatic range of the chosen microtonal grid.
<code>bach.mean</code>	Find the mean of the numbers in an llll Finds the mean of the numbers in an llll.
<code>bach.median</code>	Find the median of the numbers in an llll Finds the median of the elements in a flat llll.
<code>bach.minfo</code>	Report information about a matrix Outputs the number of rows and columns of an incoming matrix, plus an additional information
<code>bach.minimum</code>	Output minimum element-wise Compares every element of the left llll with the corresponding
<code>bach.minmax</code>	Minimum and maximum of an llll Returns the minimum and maximum numbers in an llll.
<code>bach.mod-</code>	Modular arithmetic subtraction Performs subtractions modulo a certain integer.

<code>bach.mod</code>	Quick llll modulo Outputs the positive remainders in a division operation
<code>bach.mod*</code>	Modular arithmetic multiplication Performs multiplications modulo a certain integer.
<code>bach.mod+</code>	Modular arithmetic addition Performs additions modulo a certain integer.
<code>bach.modtimes</code>	Modular arithmetic multiplication Performs multiplications modulo a certain integer.
<code>bach.mono</code>	Make bach.roll monophonic Turns a generic bach.roll (possibly containing overlapping notes)
<code>bach.mpow</code>	Compute the power of a matrix Raises the incoming square matrix to a given integer power.
<code>bach.mrandom</code>	Obtain a random matrix Builds a random matrix of given dimensions,
<code>bach.mtimes</code>	Perform matrix multiplication Performs multiplications involving matrices, vectors and scalars.
N	
<code>bach.n2mc</code>	Note names to midicents conversion Converts an llll containing note names (in latin or anglo-saxon syntax) into the corresponding llll of midicents.
<code>bach.nearest</code>	Snap numbers to nearest elements of an llll Snaps each number of an incoming llll to the
<code>bach.neq</code>	Compare two lllls for inequality Compares two lllls for inequality.
<code>bach.norm</code>	Find the norm of a vector Finds the p-norm of a plain llll. For $p = 2$ (default case),
<code>bach.normalize</code>	Normalize a vector Takes a vector and returns a new one
<code>bach.nth</code>	Return specific elements of an llll Returns one or more specific elements of an llll, based on their addresses.

O

<code>bach.omtree2bachtree</code>	Convert an OpenMusic rhythmic tree into a bach rhythmic tree Converts a rhythmic tree in OpenMusic form into the proper Measureinfo and Duration tree readable by bach.score.
<code>bach.oneperm</code>	

P**bach.p2mc****Pitches to MIDIcents conversion**

Converts an llll containing pitches into the corresponding llll of MIDIcents.

bach.pack zB one two**Assemble an llll with keyed sublists**

Assembles an llll to be parsed by bach.keys.

bach.pad**Pad, loop or trim an llll**

Pads, loops or trims an llll in order to match a given length.

bach.partition**Compute integer or set partitions**

Computes the integer partition of the incoming non-negative number or the partition of the incoming set

bach.path2llll**Obtain full directory tree in llll form**

Converts a directory path into an llll containing the full information about its files and subfolders.

Subfolders are represented by sublists, whose first elements are the subfolder names.

bach.pc2mc**Pitch-classes to midicents conversion**

Converts an llll containing pitch-classes into an identically

bach.pcsetinfo**Report basic information about a pitch-class set**

Reports some standard information about a pitch-class set, such as its

bach.period**Find the period of an llll**

Finds the minimal period of an incoming llll,

bach.perm**Return permutations of an llll**

Returns all or a subset of the permutations of an llll.

bach.pick 1 2**Return elements from an llll**

bach.pick returns one or more elements from an llll, according to their positions.

bach.pipe**Delay lllls.**

Delays one or more lllls by a specified amount of time.

bach.pitchpack**Make pitches from their components**

Generates pitches from their degrees, alterations and octaves.

bach.pitchunpack**Separate components of pitches**

Retrieves degrees, alterations and octaves from an llll of pitches.

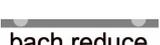
bach.playkeys cents velocity**Extract play information**

Outputs selected parameters of bach.score's and bach.roll's ployout information.

bach.poltocar**Polar to cartesian coordinate conversion for lllls**

Converts two lllls consisting of amplitudes (radii) and angle values (in radians)

<code>bach.portal</code>	<p>Let IIIs through Performs no operation upon the incoming III, except optional format (text/native) conversions.</p>
<code>bach.posef</code>	<p>Partially ordered sets Represents a lattice as a partially ordered set, both as edge list and as adjacency list.</p>
<code>bach.post</code>	<p>Print IIIs in the Max window, element by element Displays IIIs in the Max window, emphasizing their structure.</p>
<code>bach.postpend</code>	<p>Append an III at the end of input Appends a given III at the end of the input III.</p>
<code>bach.pow</code>	<p>Quick III exponentiation Raises all elements of an III (or a number) coming in first inlet,</p>
<code>bach.prepend</code>	<p>Add an III in front of input Adds a given III in front of the input III.</p>
<code>bach.prime</code>	<p>Primality test Returns 1 if the input number is prime, 0 otherwise.</p>
<code>bach.primeform</code>	<p>Compute the prime form of a pitch-class set Computes the prime form of a pitch-class set, i.e. the βmalles"copy of the set,</p>
<code>bach.primeser</code>	<p>Series of prime numbers Returns series of prime numbers, with a minimum, maximum and optional maximum numbers of items.</p>
<code>bach.print</code>	<p>Print IIIs in the Max window Displays IIIs in the Max window.</p>
<code>bach.prod</code>	<p>Multiply all numbers in an III Takes the product of all the numbers of the incoming III.</p>
<code>bach.pv public</code>	<p>Share IIIs within a patch hierarchy Shares the data it receives with other bach.pv objects with the same name, located in the same patch hierarchy.</p>
Q	
<code>bach.quantiles</code>	<p>Find quantiles or filter according to them Finds quantiles of an III or filters the III by keeping elements only inside some quantile regions.</p>
<code>bach.quantize</code>	<p>Perform quantization tasks In its simplest usage, bach.quantize converts a bach.roll into a bach.score, by using specific time signatures</p>
R	
<code>bach.r2mc</code>	<p>Frequency ratio to interval conversion Converts an III containing frequency ratios into an identically structured III</p>

	Pick random elements from an llll Outputs one or more random elements, extracted from a given pool llll.
	Output random elements with delays Outputs random elements from a pool llll repeatedly, with a given
	Compute the rank of a matrix Computes the rank of the incoming matrix.
	Display and output rational numbers Displays, inputs, and outputs rational numbers.
	Quick llll division (inlets reversed) Performs divisions between lllls or
	Read an llll from disk Reads an llll from a file.
	Read an SDIF file Reads an SDIF file and returns its contents as lllls.
	Series from recurrence relations Compute series starting from initial values and a recurrence relation
	Recursively apply binary function on elements Applies the binary function defined via the lambda loop on the first two elements, then on the previous result and third element, then on the previous result and the fourth element, and so on.
	Simplify breakpoint functions Simplifies a breakpoint function by pruning negligible points.
	Store an llll Stores an llll to be retrieved with a bang.
	Quick llll modulo Outputs the remainders in a division operation between lllls or between lllls and numbers, keeping the same sign of the left operand.
	Repeat lllls Repeats an incoming llll a certain number of times.
	Quickly find and replace elements Search and replace all occurrences of an element, or all elements satisfying a condition.
	Resample lllls Upsample or downsample any llll
	Impose to an llll the structure of another Changes the structure of an llll according to the structure of another.
	Rotate the negative signs of an llll

Accepts an **lIII** of symbolic durations, and rotates the negative signs through the elements.

bach.rev

Reverse an **lIII**

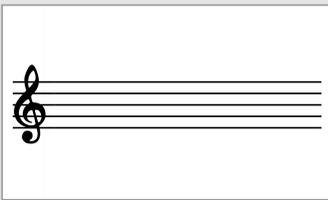
Performs a reversal of an **lIII** and its sublists, within a specified depth range.

bach.minus

Quick **lIII** subtraction (inlets reversed)

Performs subtractions between **lIII**s or

bach.roll



Display and edit a score in proportional notation

Displays a score in proportional notation, and provides the interface to interact with it.

bach.rot

Rotate an **lIII**

Performs a rotation of an **lIII** and its sublists, within a specified depth range.

bach.round

Round **lIII** elements to a value

Calculates and outputs integer multiples of **lIII** elements.

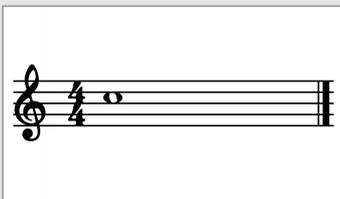
S

bach.scale

Linearly rescale an **lIII**

Maps an input range of **lIII** values to an output range.

bach.score



Display and edit a score in classical notation

Displays a score in classical notation, and provides the interface to interact with it.

bach.score2roll

Convert a **bach.score** into a **bach.roll**

Converts a **bach.score** object into

bach.scramble

Perform a random permutation of an **lIII**

Returns a random permutation of the incoming **lIII** and its sublists.

bach.shelf

Storage system for **lIII**s

Allows referring to **lIII**s by name.

bach.sieve

Only let some elements through

Retains all the elements of an **lIII** satisfying a given condition.

bach.skewness

Find the skewness of the numbers in an **lIII**

Finds the skewness (third standardized moment) of the

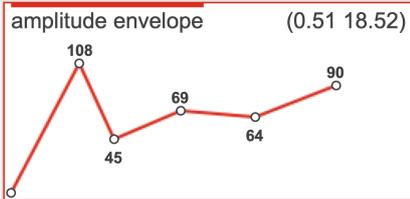
bach.slice

Split **lIII**s in two

Splits **lIII**s in two, according to a split point.

bach.sliceheader**Separate header and body of a gathered syntax**

Takes the gathered syntax of a notation object and separates the

bach.slot**Display and edit a single slot window**

Provides the display and interface for the content of a slot window.

bach.slot2curve**Convert a function-slot content into messages for curve~**

Converts the llll content of a given slot,

bach.slot2filtercoeff**Convert a dynfilter-slot content into messages to drive filtercoeff~**

Converts the llll content of a given slot,

bach.slot2function**Convert a function-slot content into a function object**

Converts the llll content of a given slot,

bach.slot2line**Convert a function-slot content into messages for line~**

Converts the llll content of a given slot,

bach.sort**Sort an llll**

Sorts an llll according to standard or custom criteria.

bach.split**Add parentheses depending on a separator**

Splits the incoming llll into chunks, depending on a given separator, and wraps each chunk inside a level of parentheses. The separators can also be specified via a lambda loop.

bach.stdev**Find the standard deviation of the numbers in an llll**

Finds the standard deviation (square root of the variance) of the numbers in an llll.

bach.step**Iterate through lllls, driven by bangs**

Traverses lllls depth-first, outputting their elements one by one as it receives bangs.

bach.stream**Stream last elements**

Makes an llll containing the last N received elements.

bach.submatrix**Get a submatrix of a matrix**

Trims and outputs a rectangular portion of a matrix.

bach.subs**Replace or remove elements in an llll**

Replaces or removes elements at specified addresses of an llll.

bach.sum**Sum all numbers in an llll**

Sums all the numbers of the incoming llll.

bach.swap**Swap elements in an llll**

Swaps two elements of an llll.

bach.symdiff

Symmetric difference of two sets
Returns the symmetric difference of two lllls.

T

bach.textin

Accept variants of the llll text format
Parses lllls according to variants of the text format.

bach.textout

Produces variants of the llll text format
Manages the attribution of backticks and the behavior with respect to negative pitches

bach.thin

Remove duplicate elements
Removes duplicate elements from an llll.

bach.tierev

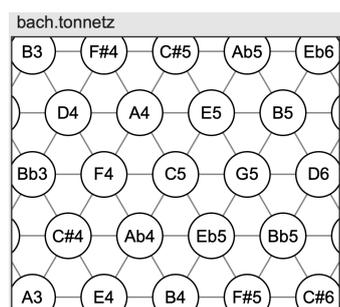
Reverse a sequence of ties
Accepts in its left inlet an llll of pitches, in its right inlet an llll ties,

bach.times

Quick llll multiplication
Performs multiplications between lllls or

bach.toc&r

Route non-rest and rest elements
Accepts an llll of elements and an llll of reference durations:



Pitch space lattice
Lattice diagram representing the (diatonic or chromatic) space of pitches,

bach.trace

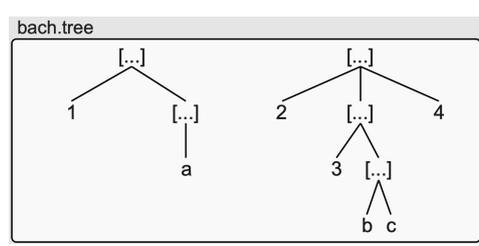
Compute the trace of a matrix
Computes the trace of a matrix, i.e. the sum of the elements on the diagonal.

bach.trans

Matrix transposition of an llll
Performs matrix transposition of an llll.

bach.transcribe

Fills a bach.roll according to incoming MIDI data
The purpose of bach.transcribe is to act as a machine which, when turned on,



Display and edit lllls in a tree diagram
Provides a simple interface for displaying and editing llll as tree diagrams.

U

`bach.union`**Union of two sets**

Returns the union of two IIIs.

`bach.unpacknote`**Separate note parameters from playout syntax**Separates the note parameters coming from the playout of a `bach.roll` or `bach.score`.`bach.urn`**Pick random elements from an III without duplicates**

Outputs one or more of random elements, extracted from a given urn III,

V

`bach.value Michael`**Share IIIs**Shares the data it receives with other `bach.value` objects with the same name, as well as bell variables.`bach.variance`**Find the variance of the numbers in an III**

Finds the variance (square of the standard deviation) of the numbers in an III.

W

`bach.weights`**Interpolate IIIs according to weights**

Interpolate among multiple IIIs according to weights.

`bach.wellshape`**Force matrix to be properly rectangular**

Pads the rows of an incoming matrix, in order to

`bach.worldcloud`

Beethoven Mozart
Brahms Machaut
Desprez Spontini Iommelli
Schumann
Stravinsky Lachenmann

Display a word cloud

Displays an interactive word cloud.t

`bach.wrandom`**Pick weighted random elements from an III**

Outputs one or more of random elements, extracted from a

`bach.wrap`**Wrap an III in parentheses**

Raises the depth of an III by wrapping it in parentheses.

`bach.write`**Write an III to disk**

Writes an III to a file.

`bach.writesdif`**Write an SDIF file**

Writes an SDIF file starting from its III representation.

`bach.wurn`**Pick weighted random elements from an III without duplicates**

Outputs one or more random elements, extracted from a given urn III,

X**bach.x2dx****Take differences between element**

Returns the differences between each element's immediate

Z**bach.zsearch****Search for Z-related pitch-class sets**

Performs a search for families of pitch-class sets having the

Operatoren**bach.-****Quick III subtraction**

Performs subtractions between IIIs or

bach.!-**Quick III subtraction (inlets reversed)**

Performs subtractions between IIIs or

bach.!/**Quick III divisions (inlets reversed)**

Performs divisions between IIIs or

bach.!=**Compare two IIIs for inequality**

Compare two IIIs for inequality

bach.***Quick III multiplication**

Performs multiplications between IIIs or

bach./**Quick III division**

Performs divisions between IIIs or

bach.%**Quick III modulo**

Outputs the remainders in a division operation between IIIs or between IIIs and numbers, keeping the same sign of the left operand.

bach.+**Quick III addition**

Performs additions between IIIs or

bach.<**Compare two IIIs for less than condition**

Compares two IIIs for less than condition, in numerical/lexicographical order.

bach.<=**Compare two IIIs for less than or equal condition**

Compare two IIIs for less than or equal condition, in numerical/lexicographical order.

bach.==**Compare two IIIs for equality**

Compare two IIIs for equality

bach.>**Compare two IIIs for greater than condition**

Compares two IIIs for greater than condition, in numerical/lexicographical order.

bach.>=**Compare two IIIs for greater than or equal condition**

Compare two IIIs for greater than or equal condition, in numerical/lexicographical order.