

# GenoMus grammar specifications

---

## Definitions

---

- **parameter**: Normalized float from interval [0, 1] used to feed the functions' required arguments. These numeric values are leaves of the functional tree that gives rise to a genotype.
- **chord**: Simplest sonic event. In traditional music writing, a note. The default attributes of a chord are: [duration, pitch, articulation, intensity]. A chord can contain more than one pitch. Chords with more parameters can be set, for example, for electroacoustic works, where a sonic event could be defined by dozens or features. A note can consist of two or more concatenated pitches together. Chords can be concatenated only sequentially (one after another, without overlapping). Overlapping effect among chords inside the same voice can be achieved by means of articulation. A chord can be extended to other domains beyond music, like visuals, lighting, etc., along with musical events, or standalone.
- **voice**: Line of music (usually for an only instrument). A voice is a wrapper for a sequence of one or more chords. A voice can consist of two or more concatenated voices together. Voices can be concatenated only sequentially (one after another, without overlapping). Each chord inside a voice can contain more than one pitch. For instance, a sequence of major chords can be a single voice.
- **score**: Excerpt or a whole piece of music. A score is a wrapper for one or more voices. A score can consist of two or more scores together. Scores can be concatenated sequentially (one after another) or simultaneously (sounding together). The product of these concatenations is a new score.

Note: The chord-voice-score structure is defined after [bach](#) paradigm, to facilitate the visualization and interactions with phenotypes in Max environment.

- **function type identifier**: Prefix used to name functions, to ease the identification of their type.

## Function types for genotypes

---

Functions in GenoMus are classified by their output data.

### Used function type identifiers

~~a b c d e f g h i~~ j k ~l m n o p q r s t~~ u ~~v w~~ x y z

### main structures

- **scoreF** (s) - outputs a score
- **voiceF** (v) - outputs a voice
- **chordF** (c) - outputs a chord
- **paramF** (p) - outputs a parameter
- **leaf** - terminal node; this not really a function type, but a tag to indicate that no new function will be called

for manual editing of leaf parameters

Function types created to allow a more user-friendly handling of decoded genotypes.

## Time

- **noteF** (n) - output normalized duration, from usual number used for note figures (1 = whole note)
- **durationF** (d) - output normalized duration, from seconds

## Pitch

- **midipitchF** (m) - output normalized pitch from standard MIDI pitch
- **frequencyF** (f) - output normalized pitch from frequency in hertz

## Articulation

- **articulationF** (a) - output normalized relative articulation (1 is whole duration of the note, 2 is double duration)
- **durationF** (d) - output normalized duration, from seconds

## Dynamics

- **intensityF** (i) - output norm. dynamics

## for outputs of an specific format

Function types created to manage specific types of data. **listF** (l) - output a normalized list of floats from interval [0, 1] - **operationF** (o) - output the result of an arithmetic operation, useful to construct recursive mathematical expressions inside a genotype - **binaryF** (b) - output a boolean value (only 0 or 1) **harmonyF** (h) - output a pitch class set, useful for specifying scales, modes, chords, pitch aggregates, harmonic series, etc. - **rhythmF** (r) - output a rythmical patter class set - **quantizF** (q) - output a numeric structure for quantization of rhythm - **externalF** (e) - output a reference from the external genotypes library (to be used with function referencing to external data - **genotypeF** (g) - output a raw encoded genotype (array of floats from interval [0, 1]) - **txtF** (t) - output a string with the path to an external txt file with data **waveF** (w) - output an encoded path to read data from an stored audio file

## Manual function for genotypes

---

Functions to be used typing manually on the genotypes to get a more controlled evolution **evolve** (*expr*) - contains the part of a genotype to be transformed, freezing the rest - **freeze** (*expr*) - block a part of a genotype to be preserved from transformations - **externalGenotype** (*genotype\_index*) - import a genotypic expression from an external file - **flatten** (*expr*) - render part of a genotype into a simple declarative score/voice/chord (phenotype)

## Indexing a function into the GenoMus catalogue

---

A functions must be included into the JSON GenoMus catalogue following this format:

```

"functionType": {
  "functionName": {
    "arguments": ["functionType", "functionType", ...],
    "description": "String describing what the function does.",
    "metadata": {
      "tag1": "Optional information when needed.",
      "tag2": "Interesting metadata could be user, date of inclusion, etc."
    }
  },
},

```

Where - *functionName* should use the corresponding prefix to help a human to know its output type - The *description* and *metadata* fields are optional, but very helpful to understand quickly the processes in a genotype, and to add further informations and documentation about the function or the programmer

Furthermore, each function must be included in the catalogue list of function index numbers:

```

"functionIndex": {
  1: {
    "functionName": 0.6180339887498949
  },
  2: {
    "functionName": 0.2360679774997898
  },
  ...
}

```

The integer index identifies the function. The float is the number to map the function on the normalized [0, 1]. Encoded genotypes uses this float to map all functions optimizing the maximal distance among functions into the interval. To maintain coherence of the catalogue, once numbered, a function can never use a different index. An index number never can be shared with other functions. The floats are generated using a golden angle iteration mapped on the [0, 1] interval using this formula:

$$\text{encodedFunctionIndex}(n) = (\text{integerFunctionIndex} * ((1 + \sqrt{5})/2)) \% 1$$