

# Mutable Instruments — Branches

---

- [Manual PDF](#)

---

[Manual PDF / Online Manual](#)

## Mutable Instruments Branches — using it for melodic patching

---

Branches is a **dual Bernoulli gate**. In plain musical terms, it takes an incoming trigger and **probabilistically sends it to one of two outputs**. That makes it great for adding **variation, phrase branching, note selection, rhythmic melody changes, and semi-random structure**.

Since this PDF is only for **Branches**, I'll focus on how its **two internal Bernoulli sections** can be used to build melodic material.

### What the module does

---

Each section has:

- **IN**: trigger input
- **CV input**: modulates probability
- **OUT A / OUT B**: trigger goes to one or the other
- **Probability knob**: biases the odds
- **Toggle mode**
- **Latch mode**

Important details from the manual:

- Output trigger/gate level is **+5V**
- There are **two identical sections**

- **Section 1 is internally routed to section 2** if nothing is patched into section 2's input
- At extreme probability settings, it behaves more like a **voltage-controlled switch** than a random router
- **Toggle mode** changes the logic from "A vs B randomly each hit" to "stay on same output vs switch to the other output"
- **Latch mode** keeps an output high until the other output is activated

## Big picture: why Branches is useful for melody

---

Branches does **not generate pitch CV by itself**. Instead, it generates **decisions**. In melodic systems, those decisions are incredibly useful because they can determine:

- which sequencer advances
- which envelope fires
- which pitch source is sampled
- which quantizer input is used
- which phrase gets played
- when a melody repeats or diverges
- when accents or ornament notes happen

So Branches is best thought of as a **melody structure and phrase variation module**.

---

## Melodic patch ideas

---

### 1. Probabilistic phrase selector

---

Use Branches to choose between **two melodic phrases**.

## Patch

- Send a clock or trigger stream into **Branches IN**
- Patch **OUT A** to trigger/clock **Sequencer A**
- Patch **OUT B** to trigger/clock **Sequencer B**
- Send both sequencers' pitch CV to:
  - separate voices, or
  - a switch/mixer/quantizer downstream for a single voice

## Result

Each incoming trigger decides whether phrase A or phrase B plays. By adjusting the probability, you can make:

- one phrase dominant with occasional deviations
- a balanced call-and-response
- rare fills from the second phrase

## Musical use

This is ideal for: - verse variation - generative lead lines - bassline evolution  
- occasional alternate cadences

---

## 2. One clock, two note lanes

---

Instead of using two independent sequencers, use Branches to split a trigger stream to **two sample-and-hold or quantizer events**.

## Patch

- Main rhythm trigger into **IN**
- **OUT A** triggers one pitch event source
- **OUT B** triggers another pitch event source
- For example:
  - A = chord tones
  - B = passing tones or tension notes

## Result

Your melody alternates unpredictably between “stable” and “color” notes.

## Musical use

This is an easy way to make melodies feel: - less looped - more human - harmonically alive

---

## 3. Ornament generator

---

Use one branch for the main melody, the other for embellishments.

## Patch

- Clock or note trigger into Branches
- **OUT A** goes to your main melodic envelope/gate
- **OUT B** goes to:
  - a higher-pitched voice
  - a short envelope for grace notes
  - a second sequencer with trills or fills

## Result

Most notes are normal, but some are replaced or accompanied by embellishments.

## Musical use

Great for: - acid-style flourishes - melodic fills - percussive synth ornamentation - occasional octave jumps

---

## 4. Controlled melodic randomness with CV over probability

---

The probability parameter has a **CV input**, which is where Branches becomes especially musical.

### Patch

- Send a slow LFO, envelope, sequencer row, or random CV into the **probability CV**
- Feed regular triggers into **IN**

### Result

The odds of choosing A or B change over time. That means your melody can slowly drift from one phrase to another, or become more/less active in certain note lanes.

### Musical use

Examples: - during a buildup, increase probability of “fill” notes - during breakdowns, bias toward sparse notes - use a bar-long envelope so phrase switching evolves across the measure

This is one of the strongest uses of Branches for melodic composition: **macro control over variation density.**

---

## 5. Cascaded melodic decisions using both sections

---

Because **section 1 is internally connected to section 2** unless section 2 has its own input, you can create **multi-stage probabilistic routing.**

### Patch

- Send one trigger stream into **section 1 input**

- Leave **section 2 input unpatched**
- Now section 1 and 2 can act like a chain of decisions

## Result

You can derive more nuanced event trees, such as: - first decision: main note or alternate note - second decision: if an alternate path occurs, choose between two different variants

## Musical use

This lets you build **3-way or 4-way melodic branching structures** when combined with additional routing or voice destinations.

For example: - Section 1 chooses “core phrase” vs “variation” - Section 2 further divides the variation into “upward fill” vs “downward fill”

That’s powerful for generative melodies that still sound composed.

---

## 6. Voltage-controlled switch behavior for deterministic arrangement changes

---

The manual notes that at **extreme settings**, the output is no longer random and behaves like a **voltage-controlled switch**.

## Patch

- Send a trigger stream into Branches
- Use CV to push probability hard left or right at certain moments

## Result

Branches becomes a performance-friendly phrase switcher: - section A during one part of the song - section B during another part - probabilistic in-between during transitions

## Musical use

This is excellent for: - moving between verse and chorus patterns - transitioning from stable melody to generative melody - live improvisation with controlled uncertainty

---

## 7. Toggle mode for melodic continuity

---

In normal mode, every incoming trigger is a fresh coin toss.

In **toggle mode**, the coin toss instead decides: - **stay on the same output** - **switch to the opposite output**

This makes the output behavior much more **phrase-like**.

### Why this matters for melody

Instead of isolated random note decisions, you get **runs** and **contiguous sections**: - a phrase may remain on melody A for a while - then switch to melody B - then stay there until another switch event happens

### Patch idea

- Enable **toggle mode**
- Use **OUT A** and **OUT B** to drive two different melodic sequencers or two transposition sources

### Result

You get larger-scale melodic coherence instead of per-note randomness.

### Musical use

This is especially good for: - AB phrase alternation - bassline variation in chunks - evolving motifs - long-form generative structure

If you want melody that feels intentional rather than “coin toss every note,” **toggle mode is the secret weapon**.

---

## 8. Latch mode for drone, held notes, and sustained melodic states

---

In **latch mode**, an output stays at +5V until the other output is activated.

### Why that's musically useful

This can create: - held states - sustained note selections - continuous enable signals for one melodic path or another

### Patch ideas

- Use A and B to enable two different quantizer/scaler/transposition paths
- Use A and B to open two VCAs controlling different pitch CV sources
- Use A and B to select between two drones or sustained harmonic centers

### Result

Instead of short trigger events, Branches can act like a **state selector** for melody.

### Musical use

You can build: - melodies that stay in one tonal area until “flipped” - alternating root notes for bass drones - long-held harmonic choices with occasional changes

This is very useful when you want structure above the note-by-note level.

---

## 9. Root note vs. extension note logic

---

A very musical trick is to use Branches to decide whether a note event should be a **strong harmonic tone** or a **color tone**.

### Patch

- Trigger stream into IN
- **OUT A** triggers a pitch source limited to root/third/fifth
- **OUT B** triggers a pitch source including 7ths, 9ths, or chromatic passing notes

### Result

Most notes reinforce the harmony, while some add tension.

### Musical use

This creates melodies that sound: - tonal but not boring - generative but still musical - expressive without needing a complicated sequencer

---

## 10. Bassline and lead derived from one source

---

Use the two sections to derive related but non-identical melodic layers.

### Patch

- Feed a master trigger stream into section 1
- Use section 1 outputs for bassline decisions
- Let the internal routing feed section 2
- Use section 2 outputs for lead or counter-melody events

## **Result**

The lead and bass become statistically related because they originate from the same incoming rhythmic source, but they won't always fire together.

## **Musical use**

This is a strong approach for: - ambient generative systems - Berlin-school style evolving sequences - modular techno with related voices

---

# **Best musical behaviors from each mode**

---

## **Normal mode**

---

Best for: - per-note random variation - fills - ornaments - alternate note triggers

## **Toggle mode**

---

Best for: - phrase-level alternation - AB structures - longer melodic continuity - motif switching

## **Latch mode**

---

Best for: - held harmonic states - drones - switching between tonal centers - sustained melodic routing

---

# Practical melodic strategies

---

## Keep one side musically safe

---

A strong patching principle: - **A = stable musical choice** - **B = more unusual choice**

For example: - A = root-position phrase - B = transposed phrase - A = diatonic notes - B = wider interval jumps - A = bass note - B = octave leap or accent

Then tune probability so the piece stays coherent.

## Modulate probability slowly

---

Fast probability modulation can be chaotic. For melodic use, slow modulation often works better: - bar-length envelopes - slow LFOs - sequencer rows changing every 4 or 8 steps

This gives evolving form rather than jitter.

## Use toggle mode when you want memory

---

Melodies often sound better when choices persist for a while. Toggle mode adds that persistence.

## Use extremes for arrangement control

---

Don't forget Branches can be nearly deterministic. That makes it useful not only for randomness, but for **composed switching** during a performance.

---

# Example melodic patches

---

## Patch 1: Evolving two-phrase melody

---

- Clock to Branches IN
- Toggle mode ON
- OUT A clocks Sequencer A
- OUT B clocks Sequencer B
- Both sequencers tuned to same scale
- Slow LFO into probability CV

**Sound:** long stretches of one phrase, with gradual transitions to another.

## Patch 2: Bassline with occasional fill notes

---

- Steady 16th or 8th-note triggers to IN
- OUT A triggers main bass pitch source
- OUT B triggers alternate higher fill pitch
- Probability mostly biased toward A

**Sound:** solid bassline with occasional melodic deviations.

## Patch 3: Harmonic state selector

---

- Latch mode ON
- OUT A opens VCA for one transposition CV
- OUT B opens VCA for another transposition CV
- Trigger Branches from a slower clock division

**Sound:** melody remains in one transposition/harmonic state until the next change.

## Patch 4: Countermelody from shared source

---

- Main trigger stream into section 1
- Use section 1 for primary voice routing

- Use section 2 as a second probabilistic layer
- Different probability settings on each section

**Sound:** two related melodic voices with intertwined variation.

---

## Bottom line

---

Mutable Instruments **Branches** is not a pitch generator, but it is extremely effective at creating **melodic structure** by making decisions about **when**, **which path**, and **how often** notes or phrases occur.

Its most melodic strengths are:

- **probabilistic phrase selection**
- **alternate note routing**
- **ornament/fill generation**
- **A/B motif switching**
- **stateful melodic variation with toggle mode**
- **held harmonic states with latch mode**
- **CV-controlled evolution over time**

If you pair Branches with sequencers, quantizers, sample-and-holds, precision adders, switches, or even just two different triggerable pitch sources, it becomes a very musical tool for building melodies that are **alive, varied, and still intentional**.

[Generated With Eurorack Processor](#)