13th International Conference on Digital Libraries for Musicology
2026, Thessaloniki. Greece

Reimagining and re-enacting polyphony (REIMAGE)
Swiss National Science Foundation (Grant 10004439)

Computational approaches to medieval polyphony: Tools and workflows developed within REIMAGE

J.-Antonio Constenla | jose-antonio.constenla-martinez@hesge.ch – **Niels Berentsen** | niels.berentsen@hesge.ch

Context

Origins

Medieval polyphony poses challenges for computational musicology:

- Fragmented sources: incomplete, or ambiguously notated manuscripts
- Theory-practice gap: theorists give rules, but composers rarely use these strictly
- Small, heterogeneous corpora : this poses a problem for constructing control groups

REIMAGE uses **fragmentology**, **digital image restoration** (MSI), **reconstruction**, and **re-enactment** for new approaches to late medieval polyphony.

Objectives

We develop computational approaches to bridge theory and practice:

- **Systematic analysis:** python/music21 tools for small corpora
- **Testable constraints:** historical rules transformed into computational workflows
- **Reproducibility:** shared methods enable cross-study comparisons

Building on previous projects at the HEM, we extend these methods to larger datasets and more complex tasks.

Corpora

Primary Corpus : Antonio de Leno's *Regule del Contrapunto* (57 two-voice examples + theoretical rules).
→ formalized as computational constraints for systematic testing.

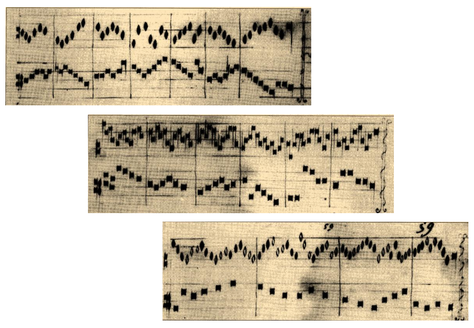
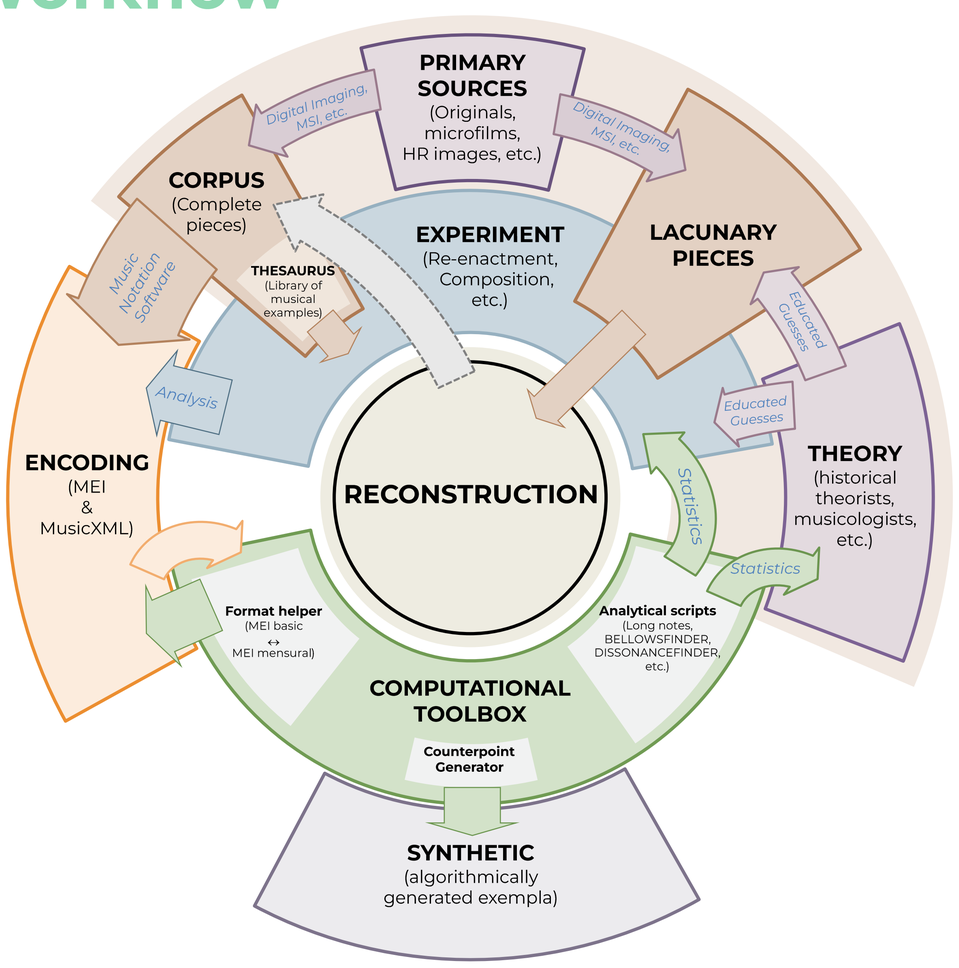


Fig. 1 – Three counterpoint examples from Antonio de Leno's *Regule del Contrapunto* (Ms. lat. Z. 336 (1581), Biblioteca Nazionale Marciana, Venice).

Reference florid corpus : 5 Trecento pieces (PMFC vol. 12 + I-Mt 1759), incl. Paolo da Firenze.

Synthetic Data : 1000+ generated two-voice counterpoints (for statistical baseline).

Workflow



REIMAGE presents a synthesis of interconnected methodologies:

- Corpus analysis produces insights useful for the reconstruction of incomplete pieces.
- **Theory and musicology** guide interpretation and constrain hypotheses, along with the examples extracted from the comparative corpus (complete pieces).
- **Experiments** test the musical plausibility of possible solutions.
- **Encoding** in MEI/MusicXML connects musicological work to computational analysis.
- **Computational tools** extract, compare, test, and generate musical material (Synthetic exempla, serving as control group).
- Reconstructions can be tested statistically against corpora of (complete) historical examples.

Computational Tools

1. Counterpoint analysis algorithms

Simultaneous long-note detection: Identifies breves or longer notes sung by all voices to analyze harmonic intervals—providing a statistically informed guide for reconstructing long notes in incomplete pieces.

Counterpoint reduction: Applies **rule-based simplification** to De Leno's florid examples to **extract the underlying simple counterpoint**.

2. BELLOWSFINDER

Tool for detecting bellows-like chains (adjacent consonances) in De Leno's exercises (*cf.* Figure 2).

- **63.2%** adjacent consonances in 1:1 counterpoint *vs.* **49.6%** in random* baseline ($p < 0.001$).
- Systematic excess over baseline for **all chain lengths**

see* **4. Counterpoint Generator

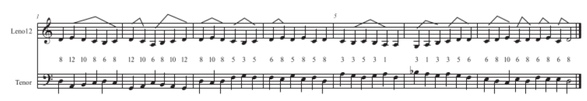


Fig. 2 – Bellows-like counterpoint chains in Leno's example 12.

3. DISSONANCEFINDER

Tool for quantifying dissonance types in two-voice counterpoint. Main types have been proposed according to melodic context (*i.e.,* **passing note**, **syncopation**, etc.) (*cf.* Figure 3).

- Middle-note dissonances (Leno's 3:1 examples): **almost exclusively regular passing notes** (Types A1–A3)
- Stepwise treatment **of off-beat dissonances** in florid examples
- Rare dissonance types limit statistical analysis in this pedagogical corpus

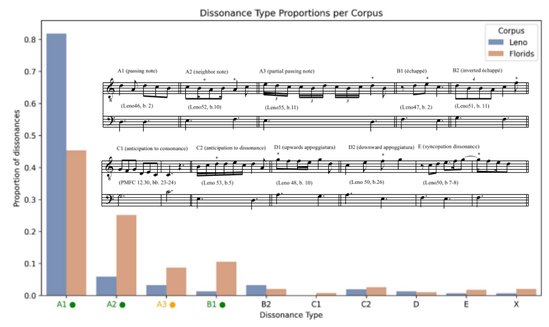


Fig. 3 – Dissonances identified by DISSONANCEFINDER & distribution of dissonance types in the studied corpora.

Statistical significance thresholds:
- p-value < 0.05 (= reliably different)
- 0.05 ≤ p-value < 0.1 (= probably different)

4. Counterpoint Generator

Generates pseudo-random upper parts on De Leno's tenors as control baseline, preserving rhythm/pitch and enforcing hard rules (perfect start/end, consonant note-against-note, etc.). Valid notes selected randomly with proximity bias (*cf.* Figure 4).

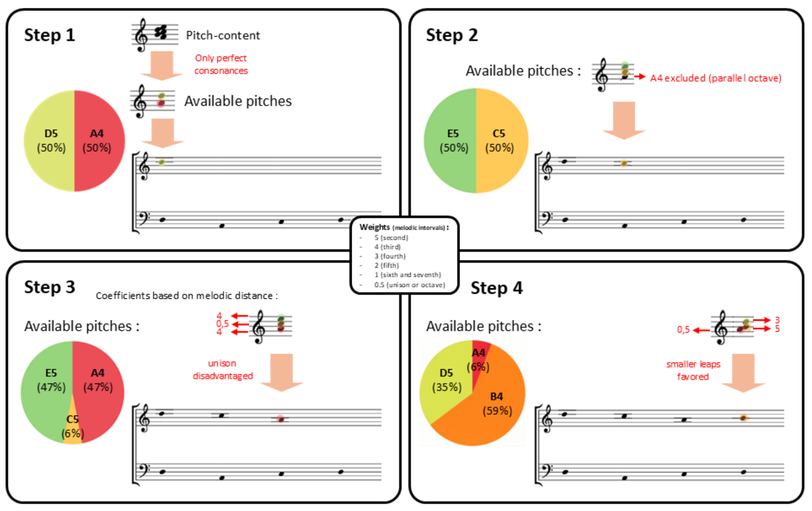


Fig. 4 – Counterpoint Generator algorithm.

Results & Perspectives

Key Achievement

- Transformed De Leno's theoretical rules into computationally testable workflows, revealing statistically significant patterns that characterize his style in a data-driven way.

Future Applications

- **Generalizable tools**: BELLOWSFINDER, DISSONANCEFINDER, Counterpoint Generator, etc. could be easily adapted to **other medieval repertoires** (*e.g.*, Trecento motets, chansons, etc.) and guide scholars in reconstruction and systematic analysis.
- **Data-driven criteria:** New metrics for **comparing** (fragmented) **sources** and identifying implicit good practices.
- **Reconstruction & re-enactment:** Informed hypotheses the reconstruction of lacunary repertoire and the re-enactment of extempore polyphony.

Schweizerischer Nationalfonds
Fonds national suisse
Fondo nazionale svizzero
Swiss National Science Foundation

hem
Genève
Neuchâtel