# Introduction to Digital Musicology:

Representing, Preserving, and Analyzing Big Musical Data

# Syllabus

## Course Description

This course will introduce you to the motivations, methodologies, and literature of computer-based music research. The class will be of interest to anyone wishing to enhance their music research skills using computers. The concepts covered are applicable to most areas of music scholarship, including performance, history, analysis, theory, composition, musicology, education, and ethnomusicology. You will be introduced to the basic computer skills and empirical analysis techniques needed to pursue computational music research, and learn how to read and critique published digital humanities research related to music. This course will focus on research with *symbolic* music data—music represented as symbols, like musical scores—as opposed to audio data.

Even as we dive into the details of computational methodology, we will maintain a critical perspective. We will ask: What are the strengths and weaknesses of digital approaches in general? What are the strengths and weaknesses of specific techniques and tools? What are the cultural, logical, and epistemological assumptions of different digital representations of music, and the tools used to analyze them?

### Learning Goals

In this class, you will learn:

- The problems and challenges of digital musicology.
- The principle techniques of digital musicology.
- The fundamentals of computational and empirical analysis.
- How to read and evaluate digital musicology literature.
- The names and oeuvres of influential, active scholars in digital musicology.
- How music is encoded digitally.
- About existing symbolic music datasets.
- About, and how to use, existing software for studying music.
- Basic computer scripting to analyze, edit, or visualize symbolic music data.
- How digital musicology can address the questions you have about music.
- What questions can't be answered by digital musicology.
- Why digital musicology is exciting and amazing.
- Why digital musicology is serious and difficult.
- What you still don't know about digital research, and what the next steps are to further your learning.

### Prerequisites

This course is intended for graduate students in music, and thus assumes extensive experience reading, performing, and writing about music, including all fundamental concepts of music theory and general knowledge of Western music history. Non-music students are welcome only if they can demonstrate sufficient familiarity and fluidity with these topics and skills. The class is suitable for students with little or no previous background in computer programming. However, all students must be fluent in basic desktop computer operation—e.g., navigating folders; creating, editing, and saving files; browsing the internet and downloading files; installing software.

# Course Structure

This course has two parallel tracks: 1) theoretical lecture and discussion, and 2) practical, hands-on, computer work. The detailed course schedule and all course materials (assignments, readings, etc.) are posted on the course website.

In the first half of the term, the theoretical discussion will focus on the methodological, epistemological, and practical issues surrounding symbolic music *data*. We will explore and learn about symbolic music representations, including MusicXML and MIDI, but focusing on *Humdrum* data. Hands-on work will focus on command line basics, text editing/searchings/formatting, and the *Humdrum* toolkit.

In the second half of the term the discussion will focus on the analysis of large symbolic music datasets, including the theoretical/philosophical fundamentals of empirical methodology, as well as specific data analysis techniques. Hands-on work will delve into basic scripting (in R), for the purposes of data analysis and visualization.

## Computer Skills

This course is not an in depth introduction to computer programming. However, students will learn the basics of navigating their computer and running simple commands through a terminal (i.e., command line), as well as simple scripting in a high-level interpreted programming language (R). Throughout the course we will work extensively with *Unix*-style terminals. Unix terminals are immediately available on any Mac or Linux operating system; students who wish to use Windows will need to install the *cygwin* Linux emulator (`www.cygwin.com/`).

All course assignments will have simple computer programming components, with concepts and skills cumulatively added over the course of the semester. Through the study of digital music symbolic representations, we will learn the basics of text editing, text searching via *regular expressions*, and text markup. Students will submit all assignments as (simple) HTML and R markdown documents (detailed instructions will be given). Through our study of data analysis and visualization, students will learn simple scripting concepts (variables, data types, functions, etc.). Students will perform simple analytical tasks (making tables, calculating correlations, etc.) and create simple graphs in the R programming language.

**Required Software:** Several pieces of software are required to participate in this class. All required software is free and open-source. If you need any help downloading or installing software, let me know as soon as possible.

**Humdrum:** The *Humdrum Toolkit* (`www.humdrum.org`) is a suite of command-line tools used to parse and analyze musical data encoded in the *Humdrum Syntax*. Humdrum is the oldest and best established framework for comprehensive digital musicology research, and will be our gateway into the world of digital musicology.

**R:** *R* (`cran.r-project.org/`) is a programming language designed specifically for data analysis. *R* includes a large suite of built in data visualization and analysis functions, making it ideal for a quick introduction to computer programming for the purposes of data analysis. *R* is easiest to use in conjunction with the software *Rstudio* (`www.rstudio.com/`), which provides a wealth of handy features for writing R code, in particular GUI interfaces which will help beginner programmers.

**A text editor:**
All assignments will be submitted as plain text files, so traditional word processors (i.e., Microsoft Word) will not serve us. Instead, we will work in simple text editor software (which are preinstalled on all operating systems). Any text editor software will do, though I recommend more sophisticated editors like *Atom* (`atom.io`) or *vim* (`www.vim.org`).

## Readings

Readings will be assigned each week. In the first half of the semester, readings will be largely drawn from the *Humdrum User Guide* (`www.humdrum.org/guide`). The Humdrum User Guide is not just a guide to the description and the use of Humdrum, but to digital musicology research methodologies in general—specific excerpts from the guide will be required, but the whole guide is recommended reading. We will also read published digital musicology literature, including corpus research papers, proceedings papers describing datasets, and relevant theoretical/methodological/philosophical papers.

## Assignments

### Homework

Simple programming/scripting assignments will be assigned each weak. The final product of these assignments will typically be no more that 10–20 lines of code, submitted as simple .txt, .sh, or .R text files.

Four short writing assignments will also be assigned over the course of the semester. These assignments will be submitted as HTML markdown documents (.md)—their length will be roughly equivalent to 1–2 pages of text in a traditional word processor. The four writing assignments will involve summarizing and critiquing parts of published articles relevant to the course topic of the week. Each student will select articles from a list provided by the instructor. Homework assignments will be due by the start of class in the week indicated in the syllabus.

**Homework presentations:** Each student will present two of their article summaries to the class in a short presentation of no more than five minutes. (Their markdown document will serve as their presentation slides.) The presentation schedule will be decided in the first week of class.

### Midterm assignment

The midterm assignment will be a proposed plan for the creation of a new symbolic music dataset, including the motivation for this dataset (who would want this data, and why?), and the methodological details of its formatting and creation (What format? What musical information? How is it encoded? How is it error checked?). (A detailed template for the paper will be provided.) The paper will be submitted as a markdown document, and should be of equivalent length to 5–6 pages in a word processor. A basic outline of the proposal must be approved by the instructor by the end of week 6. The midterm itself is due by 11:59pm on Sunday of week eight.

### Final assignment

The final assignment will be a hypothesis-driven analysis of an existing humdrum-encoded musical dataset, including both prose writing and scripting components. The prose-writing will describe a theoretical question, a specific hypothesis to attempt to address (some facet) of this question, and a ***simple*** experiment that can be carried out to test this hypothesis using the humdrum/R tools we've learning in the course of the class. The necessary humdrum/R scripts to conduct the analysis will be included as part of the final project, as well as table and graphs outputted by the scripts. The paper will be submitted as a markdown document, including prose equivalent to 5–6 pages in a word processor as well as the complete humdrum/R scripts (likely 50–100 lines of code). A one-paragraph proposal (including the desired data set, hypothesis, and analysis plan) must be submitted to the instructor by the end of week 11. The final is due by the last day of finals week. Students will also give a brief (<= 10 minute) presentation of their project and its results.

**Group final option:** Students may propose to work together on their final assignment in groups of 2–3. The scope/breadth of the final will increase depending on the number of group members: regarding the prose component, the length should be equivalent to 5–6 pages for one student; 7–8 pages for two students; or 9–10 pages for three students. The breadth, scale, and complexity of the scripting and analysis should increase commensurately with group size.

I highly encourage collaborative work, as it is an important part of empirical research; The overall work load for students working in a group should be slightly lower than for students working alone. Students working in groups will be asked to submit a short (two sentences per group member) evaluation of their own, and each of their group-mates', contribution to the group work.

## Class Forum

The class website includes an online web forum. Students are expected to participate in the forum as part of their grade. For each reading, students are expected to post at least one question on the forum, and attempt to answer (even speculatively) one of their fellow students' questions, **before the class when the reading is due**. Students are also encouraged to discuss and offer each other help with weekly scripting assignments via the forum—sharing code via the forum is also encouraged, excluding specific answers to questions asked in that week's scripting assignment.

## Attendance, Participation

Students are expected to attend and actively participate in all class sessions. Failure to regularly participate in class *will affect your grade*. Students also earn participation points by engaging with the class forum—some participation in both class and the forum is required to get a perfect participation score. Exceptional, constructive and helpful participation in class or the online forum may be rewarded with as many as five extra credit points.

## Grading

| | |
|---|---|
| In-class participation | 10 |
| Forum participation | 5 |
| Possible extra credit for participation | 5 |
| Weekly scripting | 10 |
| Four writing assignments | 20 |
| Two homework presentations | 10 |
| Midterm | 20 |
| Final | 20 |
| Final presentation | 5 |

## Class Schedule

| **Major Topic** | | **`Computer Skills`** |
|---|---|---|
| *Minor topic* | | `concepts` |
| Topic | | `(example commands)` |
| | Week 1 | |
| **Why Do Digital Research?** | | |
| | | **`Command Line Basics`** |
| *Motivations and philosophy* | | `navigation, commands, pipes` |
| Big Data & Empiricism | | `(ls, cd, mv, cp)` |
| Distant Reading, Mass Analysis | | |
| Preservation, Sharing, Access | | |
| | Week 2 | |
| **Symbolic Representations of Music** | | **`Editing and Sharing Code`** |
| *What to represent?* | | `Text editing, Markup` |
| Levels of measurement | | |
| "Primary parameters"? | | |
| | Week 3 | |
| *Music Notation* | | **`Text searching/substitution`** |
| Assumptions/limitations of notation | | `Regular Expressions, Tabulation` |
| Digitizing notation (Kern, MusicXML) | | `(grep, sed)` |
| | Week 4 **(HW 1 due)** | |
| *Beyond Music Notation* | | `(uniq, sort)` |
| Performance Data (MIDI) | | |
| Analytical Annotation | | |
| Vernacular and World music | | |
| | Week 5 | |
| **Data Collection and Management** | | **`Humdrum Syntax/Toolkit`** |
| *Manual Transcription/Encoding* | | `Spines, Tokens` |
| Consistency, Accuracy | | `(extract, paste)` |
| Redundant Annotation | | `(mint, hint)` |
| | Week 6 **(HW 2 due)** | |
| *Automated Transcription/Encoding* | | `Spine Paths, Multistops, Sections` |
| Optical Music Recognition | | `(rend, yank)` |
| Audio Feature Extraction | | `(timebase, ditto)` |
| *Error Checking* | | |
| | Week 7 | |
| *Metadata* | | **`Review/Practice`** |
| *Existing Repositories* | | |
| KernScores, MuseData | | |
| Yale Corpus, Classical Archives, ELVIS | | |
| | Week 8 | |
| | | **`Scripting Workshop`** |
| | **(Midterm)** | |

| Major Topic | | Computer Skills |
|---|---|---|
| *Minor topic* | | concepts |
| Topic | | (example commands) |
| **Summarizing and Visualizing Data** | | **Basic Scripting (R)** |
| *Descriptive Statistics* | | Variables, Assignment, Data types |
| Mean, Median, Mode, Variance | | Functions, Indexing |
| Distributions, Zipf's law | | (<-, [], table) |
| | Week 9 | |
| *Graphing* | | Graphing |
| Best practices | | (plot, barplot) |
| Scatter plots, Barplots | | |
| | Week 10 | |
| **Corpus Analysis** | | Logical tests, Control structures |
| *Empirical Methodology* | | (==/!=, if, for) |
| Random /Systematic sampling | | |
| Sample size | | |
| Hypothesis testing; Statistical tests | | |
| | Week 11 **(HW 3 due)** | |
| *Common Analyses* | | **Data Analysis** |
| Pattern-finding | | Humdrum |
| N-grams | | (patt, pattern) |
| Correlation | | |
| | | (context) |
| Regression, ANOVA | | R |
| Probabilistic modeling | | (cor, lm) |
| | Week 12 | |
| *Machine Learning* | | |
| Train and test paradigm | | |
| Statistical modeling | | |
| | Week 13 **(HW 4 due)** | |
| **Connections to Other Fields** | | **Practice/Final Work** |
| *Connections to Music Theory* | | |
| Computational models of music | | |
| *Connections to Music Cognition* | | |
| Implicit/Statistical learning | | |
| | Week 14 | |
| **Why Not Do Digital Research?** | | |
| *Pitfalls of Computational Research* | | |
| Overgeneralization, oversimplification | | |
| Mathematical obfuscation | | |
| Cultural/Epistemological biases | | |
| | Week 15 | |
| **Review** | | |
| *What have we learned?* | | |
| *What have we overlooked?* | | |
| | Finals **(Presentations)** | |

# Reading List


Arthur, C. (2017). Taking harmony into account. *Music Perception: An Interdisciplinary Journal*, *34*(4), 405–423.

Burgoyne, J. A. (2011). *Stochastic Processes and Database-Driven Musicology*. Doctoral thesis, McGill University.

Cannon, C. (2016). Sonata Form in the Nineteenth-Century Symphony. *Empirical Musicology Review*, *11*(2), 204–224.

Condit-Schultz, N. (2016). *MCFlow: A Digital Corpus of Rap Flow*. Ph.D. thesis, Ohio State University.

de Clercq, T., & Temperley, D. (2011). A corpus analysis of rock harmony. *Popular Music*, *30*(01), 47–70.

Hedges, T., & Rohrmeier, M. (2011). *Exploring Rameau and Beyond: A Corpus Study of Root Progression Theories*, (pp. 334–337). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hirjee, H. (2010). *Rhyme, rhythm, and rhubarb: Using probabilistic methods to analyze hip hop, poetry, and misheard lyrics*. Masters thesis, computer science, University of Waterloo.

Horn, K., & Huron, D. (2015). On the changing use of the major and minor modes 1750–1900. *Music Theory Online*, *21*(1).

Huron, D. (1993). Note-Onset Asynchrony in J. S. Bach's Two-Part Inventions. *Music Perception*, *10*(4), 435–443.

Huron, D. (1999). *Music Research Using Humdrum: A User's Guide*. Stanford, California: Center for Computer Assisted Research in the Humanities.

Huron, D. (2001). What is a Musical Feature? Fortes Analysis of Brahmss Opus 51, No. 1, Revisited. *Music Theory Online*, *7*(4).

Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press.

Mavromatis, P. (2012). Exploring the Rhythm of the Palestrina Style: A Case Study in Probabilistic Grammar Induction. *Journal of Music Theory*, *56*(2), 169–223.

Patel, A. D., & Daniele, J. (2003). An empirical comparison of rhythm in language and music. *Cognition*, *87*, B35–B45.

Salley, K., & Shanahan, D. (2016). Phrase Rhythm in Standard Jazz Repertoire: A Taxonomy and Corpus Study. *Journal of Jazz Studies*, *11*(1), 1–39.

Shanahan, D., & Broze, Y. (2012). A diachronic analysis of harmonic schemata in jazz. In *Proceedings of the 12$^{th}$ International Conference on Music Perception and Cognition and the 8$^{th}$ Triennial Conference of the European Society for the Cognitive Sciences of Music*, (pp. 909–917).

Shanahan, D., & Huron, D. (2011). Interval Size and Phrase Position: A Comparison and between German and Chinese Folksongs. *Empirical Musicology Review*, *6*(4), 187–197.

Strykowski, D. R. (2016). Text Painting, or Coincidence? Treatment of Height-Related Imagery in the Madrigals of Luca Marenzio. *Empirical Musicology Review*, *11*(2), 110–119.

Temperley, D. (2006). *Music and Probability*. Cambridge, Massachusetts: MIT Press.

Temperley, D., & Marvin, E. W. (2008). Pitch-Class Distribution and the Identification of Key. *Music Perception*, *25*(3), 193–212.

Temperley, D., Ren, I., & Duan, Z. (2017). Mediant Mixture and "Blue Notes" in Rock: An Exploratory Study. *Music Theory Online*, *23*(1).

Temperley, N., & Temperley, D. (2011). Music-Language Correlations and the "Scotch Snap". *Music Perception*, *29*(1), 51–63.

Von Hippel, P., & Huron, D. (2000). Why Do Skips Precede Reversals? The Effect of Tessitura on Melodic Structure. *Music Perception*, *18*(1), 59–85.