TISMIR

When in Rome: A Meta-corpus of Functional Harmony

DATASET

MARK GOTHAM
GIANLUCA MICCHI
NÉSTOR NÁPOLES LÓPEZ
MALCOLM SAILOR

*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

'When in Rome' brings together all human-made, computer-encoded, functional harmonic analyses of music. This amounts in total to over 2,000 analyses of 1,500 distinct works. The most obvious motivation is scale: gathering these datasets together leads to a corpus large and varied enough for tasks including machine learning for automatic analysis, composition, and classification, as well as at-scale anthology creation and more. Further benefits include bringing together a range of different *composers and genres* (previous datasets typically limit themselves to one context), and of *analytical perspectives* on those works. We offer this data in as ready-to-use and reproducible a state as possible at http://github.com/MarkGotham/When-in-Rome, with code and documentation for all tasks reported here, including corpus conversion routines and feature extraction.

CORRESPONDING AUTHOR:
Mark Gotham
Durham University, UK
mark.r.gotham@durham.ac.uk

KEYWORDS:
music; harmonic analysis; corpus; open science

TO CITE THIS ARTICLE:
Gotham, M., Micchi, G., López, N. N., and Sailor, M. (2023). When in Rome: A Meta-corpus of Functional Harmony. *Transactions of the International Society for Music Information Retrieval*, 6(1), 150–166. DOI: https://doi.org/10.5334/tismir.165

1. INTRODUCTION, MOTIVATION, MYTH-BUSTING

1.1 WHY, AND WHY NOW?

Recent years have seen more functional harmonic analysis corpora appearing more frequently. This is a positive development and it intensifies the need for some kind of framework to bring this material together. In previous work, we have introduced initial efforts at that coordination through the conversion of different syntaxes and corpora.[1] Here we report on a more holistic treatment which leads us to consider the 'When in Rome' meta-corpus and code library (hereafter 'WiR') ready for a v1.0, first stable release.

The benefits of coordination are clear, primarily:

- ease of consistent use at scale, and
- representation of multiple analytical viewpoints.

Creating-curating the WiR meta-corpus has required us to traverse the many complications that all meta-corpora (and indeed most individual corpora) necessarily have to navigate. We foreground the most pressing of those issues here at the outset (§1–2), even before the survey of what the corpus contains (§3).

1.2 'GROUND TRUTH'? OR ONLY 'WHAT THE ROMANS DO'

First, let us do away with the notion of a 'ground truth'. As the title of this meta-corpus clearly suggests, we take inspiration from the adage: 'When in Rome, do as the Romans do.' Beyond the pun, this saying also hints at what corpora of this kind can and cannot offer. Analysis corpora are problematic in several ways that make the term 'ground truth' almost never appropriate. Analysis necessarily involves a high degree of subjectivity and this comes out in the decisions made at all levels. For instance:

- Is a *functional harmonic* reading appropriate, or would chord symbols, or something else altogether be better?
- Even if a *functional* reading is appropriate, is this best expressed in *Roman Numerals* (hereafter RNs)? Alternatives include functions, and although a basic form of automatic conversion is possible these are two meaningfully distinct systems (see §5.5).
- Even if *RN analysis* is appropriate, which specific variant is best? Again, we discuss the details of conversion below and focus here on the more core, philosophical issues here.

Perhaps an even clearer expression of the fundamental question at stake here comes from Edsger Dijkstra's famous observation that:

> 'The tools we use have a profound (and devious!) influence on our thinking habits, and, therefore, on our thinking abilities.' Dijkstra (1982)

We consider Dijkstra's notion of 'tools' here relevant to all of the above considerations.

1.3 ONE META-CORPUS TO RULE THEM ALL?

In a word, no. As will hopefully be clear already, we recognise the intrinsic conditionality of this collection effort, and offer it as a pragmatic and hopefully useful resource to the state of our field as it stands.[2]

The potential detractions of curating meta-corpora are perhaps just as clear, including the facts that:

1. conversion involves changes, and sometimes even actual loss of data, and
2. the sum of 'all' corpora is constantly growing. We leave point 2 to consider alongside future plans in the outlook (§7). Conversion is a more pressing matter to address here at the outset. WiR seeks not only to *mitigate* but make a *virtue* of this. By providing links to all the source material and conversion routines used, WiR not only aids reproducibility, but also shines a light on the difficult corners: the gaps between different sources' syntaxes.

In short, we see this not as a burden but an opportunity. While it would be *convenient, technically* if everyone used the same syntax, this would be a *loss, musically*. Analysis is subjective, interpretative. As such, we should expect and even welcome differences of opinion as expressed both in the analysis itself and (at a higher level) in the syntaxes used.

So while conversion could be considered merely a necessary and rather tedious task, one can view the existence of separate corpora and syntaxes more positively in relation to Dijkstra's 'tools'. The current scale of the data limits our ability to make those comparisons formally, but that scale is growing, and there are already some suggestive signs of telling differences.[3]

1.4 ONLY FUNCTIONAL? ON DELIMITING THE REMIT

We delimit the remit of WiR to corpora *originally expressed* in some form of *functional* notation.[4] There are, of course, many other datasets of harmony expressed in other ways. Harte et al. (2005)'s standard focussing on leadsheet-style chord symbols is notable and widely used, for instance.

Some of those wider datasets include both chord and key information, and could thus be re-expressed in terms of RNs or other functional terms. Clearly, given the comments in §1.3, we consider that an unwarranted stretch beyond the scope. The expression of a functional reading is an interpretive step that can be plausibly –

but not reliably – deduced from keys and chords alone. Moreover this functional-or-otherwise gap typically aligns with other, corresponding gaps in the source materials (symbolic versus audio) and repertoires studied (broadly speaking, classical versus popular). These categorical distinctions are not always simple, though the combination of syntax, format, and repertoire usually makes a reasonably clear basis for drawing the proverbial line.

We note with enthusiasm certain complementary, recent efforts to integrate all of the above, including this WiR meta-corpus. Notable here is the 'Choco' chord corpus (Berardinis et al. (2023), https://github.com/smashub/choco). This kind of effort effectively 'raises the stakes': the gain in increased scale is counterbalanced by the corresponding detractions of handling such disparate repertoire, syntaxes, and analyses into a single system. 'Complementarity' here may come in the form of a distinction between large models for 'chords' in a general sense (ChoCo) and smaller, more domain-specific datasets for fine-tuning (WiR, or even its sub-corpora). In any case, there is always a practical limit. Even ChoCo, with its explicitly expansive agenda has a *ne plus ultra*, noting that while 'other collections providing harmonic information exist in the literature, some of them were … discarded.'

Finally on the question of repertoire range, we also limit WiR to the historical period of classical music for which functional tonal harmonic analysis is relatively unequivocally relevant. This centres (as the bulk of existing corpora do) on the so-called 'Common Practice' of Haydn, Mozart, Beethoven and others. *Included* at the extremes are works from slightly after 1900 (some art songs),[5] and slightly before 1600 (Monteverdi's third book of madrigals). *Excluded* are some even earlier works.[6]

In short, WiR occupies a middle position in the wider eco-system, focussing on an effort to bring together a clearly defined and closely related subset of all chordal corpora which can be loosely defined by the term 'functional tonality'.

2. FORMAT AND SYNTAX

The primary goal is to keep analyses in some format that's clear and consistent enough to be re-purposeable in many ways. Secondary is the question of how best to do this and naturally there are pros and cons to any specific approach. Before even discussing formats, syntax etc., one of the main considerations is the storing of analyses on- or off-score.

2.1 ON-SCORE / OFF-SCORE

WiR stores analyses *off-score*, while also providing methods for combining them (see §2.3). Separation of analysis from score brings certain advantages, including the following:

- **Source-independent.** Off-score analyses are not dependent on a specific score or edition. On-score analyses are either beholden to the particular score they annotate, or else they have to deal with the same detractions of off-score storage discussed later this section. Very significant here are the perennially complex issues of licence (see §2.5), and flexible alignment with other sources (other scores and indeed audio, §2.3).
- **Standalone, lightweight files.** Some tasks require analysis data only. It is beneficial in those cases to be able to work with the analyses directly without also having to parse scores, especially when the analysis-alone files (e.g., in `.TXT` or `.TSV` formats) are lightweight and quick to parse.

The leading standard for off-score analysis is 'Roman Text' (a.k.a. and hereafter 'RNTXT'). Introduced and defined by Tymoczko et al. (2019), this serves as the primary format for WiR. In addition to the *general* list of benefits to all off-score formats (above), RNTXT also supports certain *specific* functionality which is *facilitated* by being off-score, but not necessarily provided by all off-score formats:

- **Repeats.** The opportunity to identify *harmonically parallel* passages (not necessary the same as *identical* in the source) helps to see the bigger picture, avoid inconsistency, and notice where the repeat is *not* exact. See Figure 1 for an example.[7]
- **Notes.** Marginal commentary and asides are very valuable for a task like musical analysis. Much of what we want to express in analysis is not captured in RN syntax (discussed further in §7) and marginal notes allow analysts to share some of these more

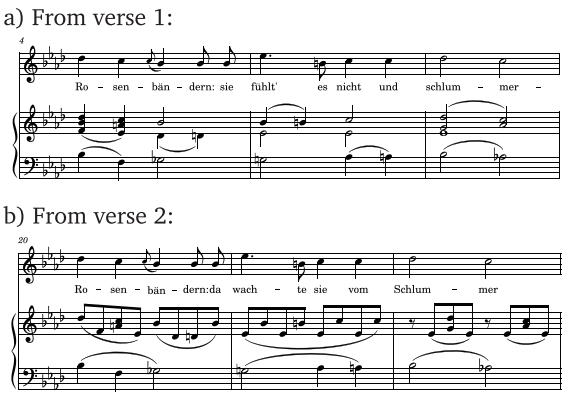


Figure 1 Two extracts from Franz Schubert's *Das Rosenband* (D.280) with varied texture but arguably the same underlying harmony. In this example, the latter case can be encoded as an harmonic repeat of the former with `m20-22 = m4-6`, or as part of larger spans, e.g., `m20-31 = m4-15`.

free-form observations. Notes often highlight the most interesting parts of the analysis and the perspective of the analyst.

The corresponding standard-bearer for on-score analysis is the DCML syntax (Neuwirth et al., 2018; Hentschel et al., 2021). Up-to-date conversion between the two is provided and discussed further in §2.4. It bears repeating that we recognise the pros and cons to each approach, that we have expended considerable effort to provide converters, and that we encourage further improvements to the ease of conversion in future. For more on this see the outlook section (§7).

2.2 OFF-SCORE, ENFORCED ALIGNMENT

With the off-score version we need tools to align source and analysis, and this is not always straightforward due to confounding factors such as anacruses, split bars, and the many kinds of annotations for repeats.

One approach is to identify mis-alignments and enforce a hard-coded solution through local ('manual') edits, though this is typically at the cost of accurately reflecting the source. For instance, this often means deleting or otherwise modifying those repeats, first/second time bars and the like.

This 'brute force' approach to alignment can certainly be justified, but typically only in 'single-use' cases where the corpus exists only to serve a specific and immediate aim. For example, 'AugmentedNet' (Nápoles López et al., 2021) is built on a version of the WiR data with alignment manually enforced. This makes sense in that case, as score-analysis alignment is clearly essential, but absolute fidelity to each source's repeat structure is not.

By contrast, a (meta-)corpus like this which seeks to serve multiple prospective research projects and interests, has different priorities. Here, it is best to keep the source data intact and make any necessary modifications only at the point of running for a specific use.

2.3 OFF-SCORE, SOFT-CODED ALIGNMENT

WiR navigates that more flexible solution with a two-pronged approach:

1. Develop the RNTXT standard and analyses in that format to encode as much of the detail for reproducing a score's information as possible. Apart from assisting with reliable alignment, the RNTXT analysis itself can be rendered as a score (e.g., in `.XML`), and so all improvements enhance the quality and usability of that analysis-score.
2. Also develop separate stream-to-stream alignment code to diagnose and solve as many of the common remaining issues as possible.

Alignment between sources of different types is one of the fundamental, ubiquitous problems not only in this micro-field, but in many corners of the wider fields of MIR and corpus study. WiR's alignment methods are handled by functionality that began life on WiR and is now available as a separate package announced and discussed in Gotham et al. (2023). By way of a brief introduction, some common cases to be relatively reliably resolved by that package include:

- **split measures:** in some sources a single measure is split into two parts, e.g., for the break between two sections. The two sources may not agree on this and so we automatically identify and resolve these cases on the analysis as needed.
- **number of measures:** as RNTXT analyses often end with the arrival of the final chord, the number of final measures is ambiguous, so we pad the analysis with final measures to match the score.
- **measure numbering:** copy the numbering from one source (e.g., score) to the other (analysis) or renumber both according to recognised conventions, e.g., starting at number 0 for an incomplete (anacrustic) first bar and 1 otherwise (when complete).
- **repeats, first/second time, and other symbols:** copy over from one source to ensure alignment/agreement.

This is not a complete list of all possible issues and the package does not provide a perfect solution to all edge cases. Nevertheless, it is a very simple, lightweight solution with a few quick checks and tweaks solving all issues encountered in the content newly added for this meta-corpus.

The 'Roman Umpire' feedback routine for score-analysis match (see §3.2.3) also helps identify possible mis-alignment: this is typically the cause of low overall-match values (below c.70%) and unusually high numbers of feedback instances. Once identified, these issues can be checked and resolved manually.

2.4 SYNTAXES (RNTXT, DCML, …) AND CONVERSION

We turn now from the 'when' conversion issues posed by mis-alignment to the corresponding 'what' issues for converting between syntaxes. Conversion code routines are crucial here and unfortunately they are rarely offered with the release of a new corpus/syntax. We appreciate and build directly on previous efforts reported to address this,[8] and we exhort creators of new standards to provide at least one method for converting to another common standard.

All WiR analyses are provided in a single format: RNTXT. This is largely because of the:

- status as an off-score format (discussed above);
- existing interoperability with music21;[9]
- number of analyses made in this format (§3).

RNTXT has been supported by music21 in some form since c.2010. Shortly after the DCML standard was first introduced (Neuwirth et al., 2018), round-trip conversion between RNTXT and DCML was added to music21 (this commit) and reported in Tymoczko et al. (2019). This converter has likewise been expanded and updated in response to DCML's much altered v.2 (Hentschel et al., 2021). In short, all existing DCML analyses have been converted to feature within WiR, and all WiR analyses can be converted to DCML with a single function. Other formats represented in the literature more occasionally are also converted here (as listed in §3's corpus overview) and supported with converters in either music21 or WiR.

In addition to these, we also offer one-directional output conversion tools to third-party apps that *visualise analyses* in alternative, user-friendly ways: Giraud et al. (2018)'s json-like Dezrann (.dez) format and TiLiA (publication forthcoming). These apps/formats support the scope for wider integration with *musical performance data* and thus unite the exclusively symbolic-time measurement in this meta-corpus with the clock-time of audio and video.

Naturally, we aspire to make conversion routines as direct as possible. In some cases, there is inevitable information loss. For example, some formats only support a limited set of chord quality options, so conversion of complex, less standard chords involved a simplified mapping (for which see §5.4). More often, the change is lossless but still affects what is *natural to express* in the language at hand (as discussed above).

In all cases, we provide 'full receipts'. In addition to conversion code, WiR also includes `remote.json` files for all external sources, with links to the source of the analysis, detailing even the specific commit wherever possible. (See also §3.2.3.) Those interested can then compare the source data with the conversion directly.

2.5 LICENCE AND REMOTE CONTENT

Licence considerations are somewhat complicated. As mentioned above, this affects even curatorial decisions such as the preference for keeping analyses off-score (§2.1). Analysts can easily write down their observations in stand-alone *off-score* files and choose to provide this as part of an open source repository under a permissive licence; to do so *on-score* requires either the analysts also encoding that score (significantly increasing the workload) or hoping that one exists and that the licence unequivocally permits their use case (rare).

There are three layers to the licence in WiR.

1. *New content* in this repository including the new analyses, new code, and the conversion of existing analyses (but not those analyses themselves) is available under the CC-BY-SA licence (a free culture licence). Exceptions may be granted, as, for example, to Sibelius, as reported here.
2. Others' analyses appear here (converted as needed) with the permission of the creators. We include clear citation-acknowledgement, sometimes with even the specific wording agreed with the originators, and with links to the source licence information where that is available. Please always refer to the source (repo. and or maintainers) for authoritative licence information.
3. Actual duplication of material is avoided wherever possible, partly as a simple good practice in handling data, and partly to simplify the licensing question. For example, the `remote.json` files can and do point to other score-only repositories, notably the Humdrum collections prepared by Craig Sapp and others and for which permission for secondary uses is unclear. This goes hand in hand with the question of source-independence, and works partly thanks to the provision for parsing scores directly from their URL.

We sincerely believe that every effort has been made to make WiR a legitimate repository with permission to exist as it does. Users considering further use cases should likewise refer to and honour the licences of all original sources.

As always, we exhort all creators of datasets (minimally) to provide clear licence information, and (preferably) to make that licence a permissive one, particularly for exact score transcription, where the task is simply to map a score into an encoded format, without any scholarly editorial intervention.[10]

From the licence perspective, the clearest part of WiR is the OpenScore Lieder Corpus (Gotham and Jonas, 2021). The score repository amounts to over 1,300 songs all freely available under the maximally permissive CC0 licence. A few hundred of those scores have been analysed for WiR (CC-By-SA). This degree of clarity and permissiveness is very rare.

3. CORPUS CONTENT AND STRUCTURE

3.1 CONTENT: AN OVERVIEW IN NUMBERS

WiR incorporates all existing, publicly-shared, encoded corpora of functional analyses. This includes new data as well as conversion from a range of sources sources and formats including:

- **RNTXT:**
  - everything reported in Tymoczko et al. (2019):
    - a few-hundred song subset of the 'OpenScore Lieder Corpus'
    - 24 Bach Preludes;
    - Baroque ground bass works by Bach and Purcell
  - expansion of those collections not previously reported (e.g., more lieder); and
  - the not-yet-published elsewhere analyses of Dmitri Tymoczko's forthcoming 'Tonality: An Owner's Manual' (expected 2023), including:

    * 371 Bach Chorales;
    * Chopin Mazurkas; and
    * Piano sonatas by Mozart (complete) and Beethoven (some).
- **DCML:**
  - Beethoven quartets (Neuwirth et al., 2018);
  - Mozart Piano sonatas (Hentschel et al., 2021); and
  - more, forthcoming material.[11]
- **Humdrum/Kern:**
  - Haydn Op.20 quartets (Nápoles López, 2017);
  - Theme and Variations movements by Haydn and Beethoven (Devaney et al., 2015); and
  - Textbook entries (Nápoles López et al., 2020),
- **Hybrid:** Sears et al. (2023)'s selection of 100 Bach Chorales is encoded in a kind of hybrid, with the *DCML* regex syntax inside a *kern* spine.
- **Other**, more one-off formats such as Chen and Su (2018)'s 'BPS-FH' corpus (32 Beethoven sonata first movements).

As this is a complex and ever-growing list, we direct users to the repository for the latest information. The `README` there lists all corpora, and the release of v1.0 is coordinated with the publication of this report for ease of reference.

At the time of writing, the corpus comprises over 2,000 analyses of 1,500 scores in the sense of individual movements: e.g., 3 or 4 for most piano sonatas and 3, 6, or 12 for many song cycles (click here for a chart on the source repository). We have more analyses than scores due to occasional overlaps, for instance, Tymoczko, DCML, and BPS-FH all provide analyses of *some* Beethoven sonata movements, none of those sets are complete and they are partially overlapping. The scores also vary considerably in length from a few cases of extreme brevity (some songs are short as 8 bars in length) to colossal movements literally 100 times longer (i.e., 800 bars).

3.2 CORPUS DIRECTORY STRUCTURE

Despite the significant variety among the target corpora, we aim to provide a logical and consistent structure overall, modelling the overall design on the lieder corpus, largely because it has (arguably) the most complex material to organise and has done so effectively. This section sets out details of that structure, and Figure 2 provides an example.

3.2.1 Overall

`<genre>/<composer>/<set>/<movement>/<files>`

- `<genre>`: A set of high-level classifications of works by approximate genre or repertoire. As most corpora are prepared in relation to this categorisation, this top level division also reflects something of the corpora's origins.[12]
- `<composer>:` the composer's name in the form `Last,_First`.
- `<set>`: extended work (e.g., a song cycle or piano sonata) where applicable. Stand-alone scores are placed in a set called `_` (i.e., a single underscore) for the sake of consistency.
- `<movement>:` name and/or number of the movement. In the case of a piano sonata, folder names are generally number-only: e.g., `1`. Most songs include both the name of the song and its position in the set (e.g., `1_Nach_Süden`)
- `<files>:` See the following sub-sections.

The 'Key Modulations and Tonicizations' corpus (Nápoles López et al., 2020) is a slight exception: we preserve the organisation of that corpus by with the `<genre>` as `Textbooks`, the `<composer>` as the textbook author, the `<set>` is the book title, and the `<movement>` is the example number, We find this exception more logical than a re-organisation by composer.

3.2.2 All folders include

- A score, either hosted locally (`score.mxl`, as discussed here) or via a link to a remote hosting (as an entry in a `remote.json` file, discussed below)
  - What: MusicXML is the most interoperable format for music notation. Our choice of the compressed (`.mxl`) version serves to minimise the file and overall corpus size.
  - How to use: Open in any notation software for (e.g., MuseScore) or notation-centred code library (e.g., music21).
- `analysis.txt`
  - What: A human analysis in plain text.
  - How to use: Open in any text editor or parse with music21's rntxt parser. These analyses can also serve as a kind of template for new work by copying the file and editing only the moments of disagreement.

3.2.3 Many folders include

- `remote.json`
  - What: this provides additional information about remote content including paths to external scores as mentioned above (§2.4).
  - This is especially important in complex cases triangulating multiple sources. See, for example, the Beethoven sonatas.
  - Additionally, we take the opportunity to provide metadata including composer name and one or more sets of catalogue information ('Opus' and/or equivalent catalogue information).
- `analysis_<analyst>.txt`
  - What: An alternative analysis from another source and/or the same analysis exactly as converted for cases of significant alteration.
  - How to use: For comparing different readings of the same work and/or as a point of reference for keeping track of the conversion process.

```
OpenScore-LiederCorpus/
|   Reichardt,_Louise/
|   |   Sechs_Lieder_von_Novalis,_Op.4/
|   |   |   1_Sehnsucht_nach_dem_Vaterlande/
|   |   |   |   analysis.txt
|   |   |   |   score.mxl
```

Figure 2 An example for part of WiR's `Corpus/` directory structure.

3.2.4 Not included but easy to generate

We include code and clear instructions for creating many additional files for any entry in the meta-corpus. The `Code/Example/` folder introduces each of these files for one example score: Clara Schumann's Lieder, Op.12, No.4, 'Liebst du um Schönheit'.

Most of the variants derive from the options for pitch class profile generation, creating files starting with the name: `profiles_` followed by:

- `<and_features_>` (optional) includes harmonic feature information. See notes at §5–6.
- `<segmentation_type>` for groupings of material by moments of change to the `chord`, `key`, or `measure`.
- `.<format>` with the options being `.arff`, `.csv`, `.json`, and `.tsv`.

Apart from these, the example folder also contains the files which are included in all folders by default (see above) as well as others that can likewise be generated across the meta-corpus:

- `analysis_<automatic_source>.txt`: Automatic analyses may be added to the dataset. Currently this is set up for AugmentedNet (Nápoles López et al., 2021): a machine learning architecture which, in turn, is built on this meta-corpus's data. This can be used in the same way as a human analysis, e.g., as a template (same format, same parsing routines). The name can be set to specify the model used, e.g., `analysis_AugmentedNet_v1.9.1.txt` in a way that is analogous to the `analysis_<analyst>` file naming for human analysts. (See also §3.3).
- `analysis_on_score.mxl`: the analysis rendered in musical notation alongside the score as an additional 'part' like that shown in Figure 3, except with the full analysis given in both RNs and function labels.
- `feedback_on_analysis.txt`: automatically generated feedback on any analysis complete with an overall rating. This is useful for proofreading. The `Code/romanUmpire.py` documentation sets out what can and can't be expected of this feedback.
- `<Keys_or_chords>_and_distributions.tsv`: pitch class distributions for each block of the score (or other source) delimited by a single key or chord as defined in the analysis.
- `slices.tsv` and `slices_with_analysis.tsv`: a tabular representation of the score in 'slices' – vertical cross-sections of the score – with one entry for each change of pitch. This is useful for various tasks, both human (at-a-glance checks) and automatic (quicker to load and process than parsing musicXML). The columns from left to right set out the:
  - `qstamp`: the time (measured in terms of 'quarter notes') since the start,
  - `measure` number,
  - `beat` number,
  - `beat_strength` deduced from the relative metrical position,
  - `Length` (also measured in quarter notes),
  - `Pitches`,
  - and (for `slices_with_analysis.tsv`), also the `Key` and `Chord` where they change.
- `template.txt`: a proto-analysis text file with only the metadata, time signatures, measures, and measure equality ranges as a template: i.e., all the information needed from the score with space to enter a new analysis from scratch.

This is too much to include for every entry. Use the example folder to see the options and to 'try before you' commit to a corpus-wide generation of one or more of these files.

3.3 AUTOMATIC ANALYSES

The WiR meta-corpus has proven useful already, as seen from the fact of it being partly or fully used in studies such as: Micchi et al. (2020); Chen and Su (2021); Nápoles López et al. (2021). It has also led to significant advance in the quality of automatic analysis as exemplified by Nápoles López et al. (2021) which is built on the data and coordination reported here. As discussed, we offer functionality for generating the output of that model (currently AugmentedNet v1.9.1) on all corpus scores. This is intended for use cases including as a:

- quick, initial survey for seeking relevant moments (see the following Anthology §4.3)
- kind of template for future, manual analyses (changing only the parts you disagree with).

On this second use case, it is reasonable to object that such 'templates' can nudge the analyst in a particular direction, encouraging them to make certain decisions. We argue that this is true to some extent in the syntax (as discussed in §1) and that the prospective gains in efficiency outweigh the detractions. In any case, any serious analyst who is discontent with the template provided will 'fix' it, or else chose to work from a blank template (which WiR also provides).

4. ADDITIONAL USES, OUTPUTS, AND INPUTS

WiR provides a great deal of supplementary code and other material, mostly in a dedicated directory (entitled simply `Code`) for implementing the functionality discussed in this paper. By showing our working, we aim to provide reasonable guarantees for reproducibility, and to encourage suggestions for improvement from users and new contributors. WiR likewise provides routines for *using* the meta-corpus in different ways.

4.1 ENGINEERING TASKS

Within the field of 'computational approaches to harmony', the lion's share of the attention seems to be directed towards *doing* automatic harmonic analysis. Naturally, at least for machine learning, this includes using analysis corpora like this one as training data. Clearly this is not the only potential use case or benefit to be had from this kind of data. In the remainder of this paper, we set out some of the wider musical use cases that are enabled by WiR. Sections §5–6 discuss the specific case of *feature extraction and classification*. Before that, we note how these analyses constitute interesting and useful data in themselves, not merely as a means to some other end.

4.2 MUSIC THEORY RESEARCH

By way of example, consider the history of music theory and the preponderance of pseudo-statistical claims about what is 'typical' of a composer or style. As the quality and quantity of datasets starts to increase, it becomes possible for current day music theory experts to assess these observations (and indeed hunches) more rigorously, and for wider groups to access that information at scale and explore it freely. Even such apparently fundamental matters like the 'rules' of voice-leading are open and active areas of research.[13]

Analysis datasets are important and useful for many of the same reasons, and perhaps doubly so given the inherent subjectivity in the notion of analytical objects like the 'chord'. While questions like the use of melodic intervals can be addressed directly from score encodings, any ideas about chords and progressions first need to extract that information from the source, either automatically, or manually.

4.3 PUBLIC-FACING ANTHOLOGY

Apart from formal research studies, it is also worth highlighting the potential for direct use in pedagogy and accessibility. For instance, see Gotham (2021) for a report of the access issues at stake in the design of a 'harmony anthology' of Open Music Theory (Gotham et al., 2021a). That anthology (click here) is built directly on the `Anthology` directory of WiR and extracted image examples for specific chords and progressions in the corpora are stored at a separate repository (here).

Many of the individual chords represented here (such as the Neapolitan) are simple enough to define and extract. Some chords, and most chord progressions are not so straightforward and require new theory building for their implementation. Discussion of these cases is well outside the scope of this report. For more detailed work on shoring up those music theoretic definitions, see Gotham (2023). By way of a brief overview, the anthology code and dataset:

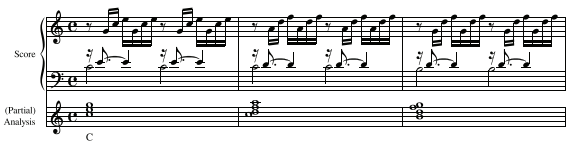
- provide frameworks for robust definitions of commonly used but under-defined terms like 'modal mixture.'
- implement concepts like the 'Common tone diminished seventh progression', which are reasonably well-defined but clearly outside the language of functional harmonic analysis and so not easy to include unambiguously in an analysis syntax.
- include a range of further, already well-defined terminology that happen to be not very widely known internationally. Despite the apparently globalised age we live in, music theoretic terms remain surprisingly regional. Examples within WiR include the 'fallender Quintanstieg' / 'aufsteigender Quintfall' pair.[14]

4.4 FLEXIBLE INPUT: PARTIAL ('SKELETON') ANALYSES

These musical, computational, and social issues also converge when it comes to the *input* of RN analysis, and the brittle syntax where a character typo can completely alter the meaning of the chord. This is challenging to both humans and computers trying to learn the subject. Human analysts, especially students, naturally struggle with this and benefit from assistance. WiR provides support by accepting partial ('skeleton') analyses as a valid input.

Figure 3 shows two extracts from the example used in WiR's test suite. Here, a minimal analysis of the first

bb.1–3: Chords pitches and one key indication



bb.20–22: two secondary tonalities:

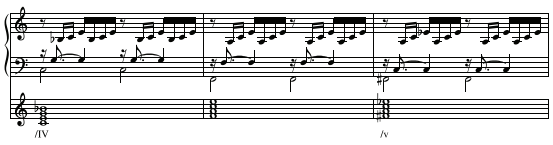


Figure 3 Extracts from the partial analysis test provided in the corpus. The two extracts show full pitch reductions but very little textual analysis. WiR fills in the blanks to complete the picture.

Bach prelude is expressed primarily in the pitches and with only a small amount of text annotation, noting the moments of key-changes and tonicisations.

This could be considered to *halve* the task of harmonic analysis from 1) thinking about what the harmonic reduction is and 2) expressing that in the jargon of RNs, to simply step 1) alone. As there are so many variants on RN syntax, even the most seasoned experts make typo-style errors, so this is labour saving not merely for the novice. The corresponding Romantext would express these two sets of three chords as:

```
m1 C: I
m2 ii2
m3 V65
...
m20 V7/IV
m21 IV7
m22 viio7/v
```

Moreover, even this may be specifying more than is necessary. First, the secondary tonalities can also be drafted automatically by the `preferSecondaryDominant` functionality which was introduced to music21 as part of the present effort (here). Additionally, Gotham et al. (2021b) suggests that if humans provide only the segment boundary information, the rest of a complete RN analysis can be deduced with surprising accuracy using only simple pitch-class profile based algorithms.

And this is not only a consideration for humans. Machine learning models are subject to a similar kind of error-tendency: many are (more) able to correctly deduce the pitch reduction aspect of a harmonic analysis, while still struggling to get the RN syntax expression exactly correct and consistent.

5. PROFILES AND FEATURES

A great deal of prior work has been devoted to feature extraction from musical sources. This includes work on both the audio and symbolic sides, and for a range of parameters both *within* music (pitch, timbre, texture, ...) and *beyond*.[15] Use cases include machine learning classification tasks: in defining a wide range of musical domain features, we can establish which of them are the most predictive of categories such as composer/artist or genre/style.

These features can be more or less effective as discriminators, and also (it is important to note) more or less useful in guiding human understanding of the musical matters. Prior work has arguably (and understandably) prioritised computational efficacy over the human explanatory side, and on readily available sources, particularly audio. Within this, a wide range of feature types has been considered for musical parameters including pitch, rhythm, and lesser-studied elements such as texture (Bigo et al., 2018).

The human harmonic analysis datasets represented in WiR may be useful as a source of features both distinct from – and in combination with – data from the corresponding (audio or symbolic) source. No set of features specifically for the information afforded by these analyses has previously been offered.[16] To be clear, existing features sometimes concern these mid-level analytical considerations, but the process of extracting the analysis is a part of the feature extraction itself rather than a pre-existing source, and it often involves a heavily simplified notion of the chord. For example, jSymbolic (McKay et al., 2018) includes chord features (C-1, C-2, ...) but operationalises the notion of a 'chord' in terms of vertical cross-sections (similar to the slice approach seen in White and Quinn (2016) and discussed above, §3.2.4).

This view of the 'chord' is perfectly understandable when everything has to be extracted automatically from the source, but as we now have a good provision of human analysis datasets to work with, and as automated harmonic analysis is rapidly getting to a level of quality where it would be meaningful to analyse those results directly, it is time to start investigating features extracted from the analyses themselves (whether human or high-quality automatic). WiR provides code for extracting features directly from data hosted there. This section (§5) provides an overview and some preliminary remarks common to many features and §6 discusses a preliminary set of specific features.

5.1 HARMONY FEATURE TYPES

Features for harmonic analysis may be of several types, including at least the following four:

1. **Each individual chord in an analysis (§6.1)**, for instance concerning the intrinsic properties of a chord (such as its 'quality'), its position in a work (measure, beat, beat strength), and corpus-contextual matters such as its relative rarity. All of this can be performed on the exact harmony asserted by the analyst or on various kinds of simplified versions (as discussed in §5.4).
2. **Two or more chords in an analysis (§6.2)** enable features including *n*-gram progression types and harmonic rhythm.
3. **Comparison of a chord and the corresponding source (§6.3)** illuminates the relationship between the two, including how completely an analysis accounts for the source.
4. **Global attributes (§6.4)** identify overall features such as the number of chords and the average and standard deviation of various local features.

5.2 PITCH CLASS PROFILE (PCP) MATCHING

Many of these features require pitch comparisons. Pitch class profile (PCP) matching is a commonly used, effective, and highly interpretable method for comparing the pitch content of two elements (whether chords or sources or both). PCPs consist of a 12-dimensional vector with one entry for each of the pitch classes (0–11). Simple prototype profiles use binary values with 1 for presence and 0 otherwise. For instance, a binary profile for the chord of C-major (C, E, G) would be given as:[17]

```
[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0].
```

Wrapped rotations of these profiles cover all the transposition-equivalent sets; in this case, all major triads (D♭, D, E♭, …).

Alternatively, profiles can be constructed from musical sources (audio, symbolic, piano roll, …), also producing a 12-dimensional PCP, but with continuous values for each entry (except in the very simplest cases). Here, passages corresponding exactly to the human-defined segmentations by chord are gathered into groups by chord type in order to build up profiles from passages more robustly asserted to be within-type.

Many prototype profiles have been offered in the literature for *keys*, and Gotham et al. (2021b) use WiR data to introduce the first corpus- and analysis-derived profiles specifically for *chords*. Whether binary, corpus-derived, or otherwise, we can compare any pair of such PCPs with standard difference measures (e.g., with ℓ^1 or ℓ^2) and examples of potentially useful comparisons for our purposes are formalised as part of §6.

5.3 CHORD TYPES

PCP mapping requires a fixed list of reference chord types. It is typical here to use a short list of 9 such types: the four triads (major, minor, diminished, augmented) and five of the most common sevenths (dominant, major, minor, diminished, half-diminished). The binary PCPs for these 9 canonical chord types are as follows when rooted on C (or, rather, on pitch class 0):

```
diminished triad:
[1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0],
minor triad:
[1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0],
major triad:
[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0],
augmented triad:
[1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0],
diminished seventh:
[1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0],
half-diminished seventh:
[1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0],
minor seventh:
[1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0],
dominant seventh:
[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0],
major seventh:
[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1]
```

Around 95% of all chords in the WiR corpus are accounted for by one of these. The remaining c.5% include:

- additional seventh types such as sevenths built on augmented triads;
- all further tertian chords (i.e., 9ths: there are no 11th or 13th chords in the corpora);
- some chromatic chords like the 63 and 43 forms of the augmented sixth (the 653 is enharmonically equivalent to a dominant seventh); and
- detailed entries specifying missing, added, and altered tones.

5.4 SIMPLIFICATION (AND 'CONSOLIDATION')

There are contexts in which we will want the exact, original chord as declared in the analysis. Equally, there are other contexts where it is more appropriate to narrow the many possibilities into a smaller set. There are two basic ways of achieving this feature dimension reduction. The first is simply to add a category ('Other') for all variants. The second (and usually more effective) way is to simplify chords until they map naturally and more evenly into the categories available. The 'Other' category may still be needed (depending on the severity of the simplifications) but will be much less heavily used.

Exactly how best to go about simplifying these chords is an intriguing question in itself which does not map to a uni-linear progression from 'most complex' to 'simplest'. It is possible to simplify specific aspects of a chord independently. Some mappings even simultaneously reduce and increase the complexity (see §5.5 below for a case in point).

Options include (in approximate order of severity):

- removing the 9th of a 9th chord to yield one of the 7th chord types above (e.g., V9→V7).
- 'completing' incomplete triads (V[no3]→V).
- removing all inversion information (V65→V7).
- reducing to function only (V→D; see §5.5)

Again, more or fewer of these simplifications can be implemented depending on the task at hand. For instance, implementing the first two only (so mapping to simple triads and sevenths, while remaining with RN labels including inversion) roughly halves the overall number of distinct entries. Figure 4 uses this simplification to show the most common chords in major key contexts of the OpenScore lieder sub-corpus. The minor case follows a comparable pattern, also starting with the root position tonic `i`, followed

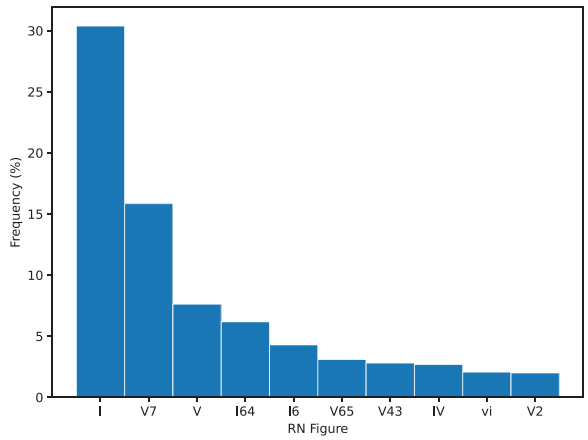


Figure 4 The 10 most common figures in major contexts of the lieder corpus, after simplification and 'consolidation' as discussed in the main text.

by `V` and `V7` (sic, now this other way round), and `i64`, `i6` and `V65`. Thereafter, the long tail values diverge more.

Code within WiR supports any combination of these simplification methods. It also provides a lighter-touch form of this method which we call 'consolidation' that does none of these chord-changing simplifications, but simply maps equivalent symbols to the same entry. For example, the third inversion of a dominant seventh chord can be annotated as 'V642', or in one of the shorthands 'V42' and 'V2'. As all three are equivalent, the consolidated method always returns the most compact form (here, 'V2'). Moreover, WiR offers what we call 'careful consolidation' in which the pitches implied before and after the change are derived, compared, and required to be unchanged for the internal mapping to go ahead. Summative data (including that used for fig.4) is available in `.json` files in the `Code/Resources/Chord_usage/` directory.

5.5 FUNCTIONS AND/AS SIMPLIFICATIONS

Functional notation (*Funktionstheorie*) is a way of describing tonal harmony that typically stands as an alternative to RNs (*Stufentheorie*). However, the two are broadly compatible, and at least the simplest forms of the Functional notation provide a clearly structured method for chord reduction in two levels. At the top-level, we have the main (*Haupt*) function only with tonic (T), subdominant (S), and dominant (D) in upper and lower case for major and minor, i.e.:

```
'T', 't', 'S', 's', 'D', 'd'.
```

At the secondary level, we add *Neben* functions to modify the *Haupt* functions with `p`, or `g`, or neither:[18]

```
'T', 'Tp', 'Tg', 't', 'tP', 'tG',
'S', 'Sp', 'Sg', 's', 'sP', 'sG',
'D', 'Dp', 'Dg', 'd', 'dP', 'dG'.
```

WiR provides a simple mapping from RNs to functions which uniformly *reduces* the information content to 6 or 18 categories. This is broadly 'correct' though it glosses over some edge cases where the functional labels ought in fact to *increase* information. Put another way, while many RNs map to a single function, some single RNs map to multiple functions. For example, in a major key, the functional system has a choice of either 'Tg' or 'Dp' for the mediant ('iii'), There is a choice, that choice is not neutral, and the application of 'iii' to 'Tg' in this system is an assumption based on what is informally attested to be 'more common'.

As all existing corpora of functional harmonic analysis are encoded in RNs, there is little we can do to nuance this for now. It would be good to see some harmonic analysis corpora built directly on functions in the future. We hope that the mapping functionality here may encourage those developments and perhaps even widen the user-base for this kind of corpus building and study.[19]

5.6 ONE-HOT ENCODING

Finally, for machine learning we will often wish to encode information 'one-hot' (with a single '1' entry among several '0's). Many of the features described here can be encoded in this way. Exceptions where this is impractical include continuous value PCPs for which there are too many possibilities to create an indexable list.[20]

6. FEATURES BY TYPE

Following jSymbolic, let us label features with tags by type, all starting with **H** for harmony (to distinguish from jSymbolic's notion of a chord as described above) and with an additional tag for each of our four groups: **'I'** for individual chords, **'P'** for chords pairs, **'C'** for source-chord comparisons, and **'G'** for global attributes. To itemise the features clearly and consistently, we will provide that tag (e.g., HC1), a name, a *discrete* or *continuous* label, the number of dimensions, and (where appropriate) a brief explanation.

6.1 INDIVIDUAL CHORDS. HI1, HI2, ...

1. **'chordQualityVector':**
   Discrete
   10-Dimensional: diminished, minor, major, and augmented triads; and diminished, half-diminished, minor, dominant, and major seventh chords; as well as `None` as discussed in §5.4.
2. **'thirdTypeVector':**
   Discrete
   3-Dimensional: minor, major, or None/other.
   For the third and fifth, 'None' means the analyst has specified a chord with no such entry (e.g., a dominant triad with no third encoded as 'V[no3]') and that this information has not been simplified

away in processing the features. This annotation type is relatively rarely used, even when it would more accurately describe the source than the corresponding 'complete' triad.

3. **'fifthTypeVector':**
   Discrete
   3-Dimensional: diminished, perfect, or None/other. (As for 'thirdTypeVector'.)
4. **'seventhTypeVector':**
   Discrete
   4-Dimensional: diminished, minor, major, or None/other (4).
   For the seventh, 'None' can mean a triad alone. Triads are extremely common, of course. The annotation of a 7th without 7th is almost never appropriate, though it is theoretically possible, e.g., in the case of 9th chord without 7th ('V9[no7]') simplified such that the 9th is lost (only triads and sevenths) but the incompleteness is retained.
5. **'rootPitchClassVector':**
   Discrete
   12-Dimensional: a pitch class from 0–11 (C–B).
6. **'hauptFunctionVector:'**
   Discrete
   6-Dimensional: As discussed in §5.5.
7. **'functionVector':**
   Discrete
   18-Dimensional: As discussed in §5.5.
8. **'chosenChordPCPVector':**
   Continuous
   12-Dimensional: As discussed in §5.2.
9. **'fullChordCommonnessVector:'**
   Continuous
   1-Dimensional: A measurement of how often this chord appears in the corpus, mapped to the range 0–1 where 1 is the value for the most commonly used chord (note, not the total).
10. **'simplifiedChordCommonnessVector':**
    Continuous
    1-Dimensional:
    As for 'fullChordCommonnessVector', but with the simplified set of chords discussed in §5.4.

Note that there is clearly redundancy here: the triad quality includes the information about 3rd, 5th, and 7th types. Nevertheless, feature analysis can be opaque: it is not clear which of these versions of the same/overlapping information will prove most effective in a given context *a priori*, without simply undertaking the analysis. WiR therefore adopts a 'more-is-more' approach.

6.2 PAIRS OF SUCCESSIVE CHORDS (BIGRAMS). HP1, HP2, …

1. **'diffChordPCPVector:'**
   Discrete
   12-Dimensional:
   This new binary PCP records the *difference* between the existing binary PCPs of the pair of chords (with 0 for matching entries by index and 1 for changes). This groups specific, *absolute* progressions, e.g., from A♭ to G♭ major triads, regardless of the functional annotation.
2. **'diffRootRotatedPCPVector:'**
   Discrete
   12-Dimensional:
   As for 'diffChordPCPVector', but rotated to the first chord's root note to account for transposition-equivalent progressions. To continue with the A♭ → G♭ example, this now captures that specific programme alongside any progression between two major triads where the second chord's root is two semitones below the first (e.g., G → F).
3. **'diffKeyRotatedPCPVector:'**
   Discrete
   12-Dimensional:
   Once again, this captures classes of transposition-equivalent progressions, but now rotating the 0 index to the key's tonic to account for only key-relative equivalent progressions. Returning to our example, this will see A♭→G♭ as equivalent to G→F only if they are expressed in similar key-relative ways, e.g., as V-IV in their respective keys (here, D♭ and C).
4. **'harmonicRhythmPair:'**
   Discrete
   5-Dimensional:
   The length of the second chord in proportion to the first. We limit options to: 3 (i.e., 3× as long), 2 (e.g., for 2+1 rhythms), 1 (equal in length), 1/2, and 1/3. For example, the one-hot vector for equal length is: [0, 0, 1, 0, 0]. Other values are filtered to the nearest in this list, so 'double dotted' rhythms (3.5:0.5) will be mapped to 'single dotted' (3:1), as will all proportions greater than 3:1.

The first three of these vectors relate to classes of chord progressions. They would support, for example, measures of transition probability: the Bayesian likelihood of a chord given the preceding one. These could, in turn, be measured against the frequency of that specific progression in a reference corpus.

The motivations are clear: transition probability is highly distinctive as chords do not follow each other with equal probability. For instance, harmonic progression is asymmetrical: for many pairs of key-relative chords, the transition probability $X \rightarrow Y$ is not equivalent to $Y \rightarrow X$. Moreover, these transition probabilities are repertoire dependent. For instance, IV-V is much more common in classical music than the reverse, but that is not true of rock music, for which V-IV is common.[21] And while V-IV is rare in a classical context, that rarity itself can be a source of musical interest. Figure 5 shows a delightful example

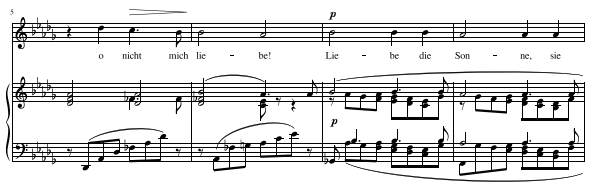


Figure 5 An extract from Clara Schumann's setting of Rückert's 'Liebst du um Schönheit', with V (A♭ major, second half of b.6) leading to IV (G♭ major, b.7) as an example of less-common figure-to-figure transition probability.

from Clara Schumann's setting of Rückert's 'Liebst du um Schönheit' with IV following V (admittedly at a phrase break).[22]

Turning to harmonic rhythm, this may be regular (1:1 ratio) in another simple proportion (e.g., consistently 2:1), or something else altogether. For example, there are cases of extremely static harmony, such as Wagner's *Rheingold* which famously begins with 136 bars of continuous E♭ major (yes, the *chord*, not merely the *key*). Figure 6 shows a more relevant and typical extract from the start of Franz Schubert's 'Trockne Blumen' (from *Die schöne Müllerin*, D.795) with an initially very static harmony (4 bars of tonic) that only gets moving in preparation for a G-major cadence (2 changes per bar in bars 5–6).

6.3 CHORD-SOURCE COMPARISON. HC1, HC2, …

Chord-source comparison can be handled by PCP matching as discussed in §5.2 above. Potentially useful comparisons can be formalised in terms of the features of the following list, many of which depend on having identified the 'best fit' chord which in itself is identified by PCP matching between the source and a set of options. All *distance vectors* are **continuous** values, mapping to the range 0–1; all *match vectors* are **discrete**. All of both are **1-Dimensional**.

1. **'distanceToFlatPCPVector':**
   Continuous
   Comparing a source passage with the flat (equal) PCP yields a value indicative of the *clarity* of a chord. Flat profiles are highly chromatic, though this does not necessarily correspond to tonal ambiguity.
2. **'distanceToChosenChordVector'**
   Continuous
   Comparing the PCPs for the source passage and the assigned harmony indicates how much of the source pitch content corresponds to within-harmony (chord) tones as opposed to non-harmonic (e.g., passing and neighbour) tones.
3. **'distanceToBestFitChordPCPVector':**
   Continuous
   By considering all chords types and roots (i.e., rotations), we can find the one that fits best from the PCP perspective. This may or may not be the one asserted in the analysis.[23] Having identified the 'best fit' chord, this feature captures the distance between it and the source, just as 'distanceToChosenChordVector' does for the analyst-asserted option.
4. **'distanceChosenToBestFitChordPCPVector':**
   Continuous
   This variant on the two features above compares the 'best fit' with the 'asserted' chord, without reference to the source. This illuminates the clarity of the choice and the difference may be slight as, for example, between V and V7 or IV and ii65.
5. **'chordTypeMatchVector':**
   Discrete
   Is the asserted chord of the same quality as the best fit?
6. **'chordRotationMatchVector':**
   Discrete
   This is a variant on 'chordTypeMatchVector', now concerning the match or otherwise between the root (or 'Rotation') of the chosen and 'best fit' chords.

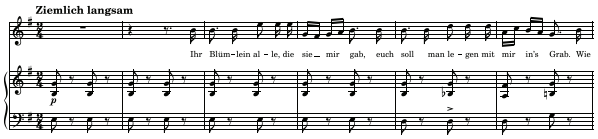


Figure 6 An extract from Franz Schubert's 'Trockne Blumen' with initially static harmony.

6.4 GLOBAL ATTRIBUTES. HG1, …, HCG1, …

Among 'global' attributes that provide an overview of a whole piece, we can distinguish between *global versions* for the previous discussed vectors, and *uniquely global* aspects.

For the previously discussed vectors, let us define a parallel set with 'G' inserted. For instance, HC1 becomes HCG1, with the former recording one chord's distance to the flat PCP profile and the latter referring to the average of all such distance entries for the work in question. Most vectors can be meaningfully summarised in this way with averages or histograms, albeit with dimension and typing changes. For instance, 1-dimensional True/False Boolean values now become continuous values across the range 0–1.

Turning to the new, specifically global, we propose four, simple counts of chords and keys:

1. **'numChords':**
   A simple count of the number of chords overall, including the occasional explicitly encoded repetition of the same chord: i.e., the number of chord changes + 1.
2. **'numDistinctChords':**
   The smaller number of *distinct* chords in the work. Both chord counts (this and 'numChords') will be affected by the extent of chord simplification employed (§5.4).

3. **'numKeys':**
   As for 'numChords'.
4. **'numDistinctKeys':**
   As for 'numDistinctKeys'.

The number of *chords* will typically be much larger than that of the *keys*, and each will be greater than the corresponding *distinct* counts. Although the counts are discrete, a continuous (1-dimensional) version for each can be computed by normalising by the maximum or total value for the corpus studied (or by some predefined threshold), resulting in a value in the range 0–1 relative to the corpus-level total.

7. OUTLOOK

This paper has set out the content of the WiR meta-corpus and code library, discussed in some detail certain possibilities and pitfalls inherent in a collection of this kind, and set out a range of use cases. We conclude with a look to the future, both of WiR specifically, and in a wider sense, of the discipline.

7.1 IT'S USEFUL AS LONG AS IT'S USEFUL

Projects like this are hard to maintain. Other corpora are now emerging more quickly than previously. Clearly this is a good thing, but it makes the task of integrating all corpora more challenging, with more regular updates required. Second, as with any project of this kind, there is the question of long-term storage preservation and management. Long-term archiving is a significant challenge for digital heritage and we do not claim to have any detailed plans for the long term.

What we can say is that we are open to options, ideas, and collaborations including prospectively porting this data elsewhere. The criteria for doing so effectively double as the list of benefits of – and motivation for – working on this meta-corpus:

- **Neutrality.** WiR is a meta-corpus of functional harmonic analyses in which the *new contributions* here are a relatively small part of the total. That alone makes it well placed to try and balance the challenges of conversion without appearing too partisan for one school, format or the like.
- **Licence.** The licence is as permissive as realistically possible for a project of this kind.

Any prospective inheritor of this data would need to address these considerations.

Equally, there is a chance that the collection may simply become superseded. This large collection of high-quality analyses could be considered obsolete in the case of (much) better automatic analysis tools. I.e., if automatic analysis were 'as good' as human analysis, the motivation would be greatly reduced. We're not there yet, but the rapid advances of the past few years may indicate that such a scenario is possible and perhaps even imminent.

'Obsolescence' would also mean non-preservation of individual analysts' views expressed here. This sounds drastic but may be reasonable if automation reached the point of creating and managing the range of different viewpoints. That is, we could take the view that there is little need to preserve individual human analyses if a computer system were able to account for *all and only the credible views* of even the most complex passages in a clear, structured, and consistent way. If our human analyses serve to help us reach that level of automation, then the job is arguably 'done' and the motivations for archiving are 'merely' out of historical interest rather than ongoing utility.

We do not view even this hypothetical future as the 'end' of musical analysis, largely because we view harmonic analysis as one aspect of a much wider, holistic approach to understanding and appreciating music. Different possible analytical readings may be compared with *each other*, with *other parameters*, and with *possible meanings*, and more besides in what amounts, clearly, to a process of enrichment. In this view, the bread-and-butter task of producing reductive summaries of chord progressions in a work is a small cog in a complex system, and one that analysts may be happy to have done 'for them'. Rather than an 'end' to musical analysis, then, this kind of automation may taken up as an invigoration of it, supporting the human involved to focussing on the higher-level tasks to which they are better suited.

7.2 THE MOVING TARGET

As for further corpora and syntaxes, clearly we welcome the addition of more data, more perspectives, more repertoire coverage, more structured observations of more differences in approaches … and more. There is a trade-off between adopting existing formats and inventing something new. As we discussed at the outset of this paper, while there is a benefit to having everything in the same format, there is also something to be said for supporting different syntaxes, especially where the new syntax offers something unique. Those tempted to devise yet another syntax should not be put off entirely but should (please):

- consider whether it offers something new, distinct, and valuable;
- provide conversion code to at least one other more established standard rather than leaving this to future meta-corpus creators. We welcome a discipline-wide discussion of these points, particularly as it pertains to the coordination and future-orientation of all kinds of analysis – not merely harmonic.

7.3 DIVERSITY

Let us give the last word, humbly, to diversity of representation in our field. The external corpora collected for WiR are extremely un-diverse, featuring *only and exclusively* music by white male composers. As such, what diversity WiR offers is the result of an in-house effort. This issue is not unique to harmonic analysis corpora, of course – it is partly a symptom of wider forces in classical music – though even extremely early proto-datasets like Barlow and Morgenstern's *A dictionary of musical themes*, include at least *some* range.

More positively, among the analyses newly prepared for WiR (i.e., not adopted from elsewhere), there is a good degree of diversity. This is thanks largely to the dedicated effort of the lieder score corpus for which a wider and more diverse range of protagonists was a key motivation, and to a corresponding dedication here to ensuring that the works chosen for analysis are at least as diverse as the original corpus. That is small proportion of the 2,000 analyses overall, but more than a drop in the ocean, and a positive start.

8. REPRODUCIBILITY

As discussed in the main text, all data, code, and more in this project is offered as freely as practically possible at: http://github.com/MarkGotham/When-in-Rome.

NOTES

1 See Tymoczko et al. (2019); Micchi et al. (2020); Nápoles López et al. (2021).

2 See the concluding section §7 for thoughts on the future.

3 See Gotham (2021) for a discussion.

4 In the practice of existing corpora, this always means RNs specifically, though again, see §5.5 for a discussion of wider possibilities.

5 The lieder corpus includes scores for works as late and 'post-tonal' as Webern but WiR does not include functional analyses of them for obvious reasons.

6 Dmitri Tymoczko's forthcoming 'Tonality: An Owner's Manual' (expected 2023 and discussed further below) analyses Palestrina, Dufay and more. We do not argue against the use of functional labels for the analysis of those early repertoires, though we do consider it more of a debatable point and out of scope here.

7 All musical examples used here can be found in the WiR meta-corpus. Most, including Figure 1, are of songs in the OpenScore Lieder Corpus (Gotham and Jonas, 2021).

8 See, for instance, Tymoczko et al. (2019) and Micchi et al. (2020).

9 First reported in Cuthbert and Ariza (2010), music21 is a well-known library for symbolic music processing with a provision for RNTXT that continues to be maintained and developed. Some of the code provided also builds on music21.

10 For a discussion-cum-proposal on this subject, see Gotham (2021).

11 Notable here is the 'Romantic Piano Corpus': publication forthcoming as Hentschel et al. 'An Annotated Corpus of Tonal Piano Music from the Long 19th Century'.

12 As discussed, every analysis and `remote.json` file includes an attribution for the avoidance of doubt.

13 See Huron (2016) for a wide-ranging overview from a major protagonist of this field.

14 See (Lewandowski, 2010) for an introduction and Gotham (2023) for an analysis of occurrence in this data.

15 For instance, see Vatolkin and McKay (2022) for an approach to handling six feature-source types together: 'audio signals, semantic tags inferred from the audio,symbolic MIDI representations, album cover images, playlist co-occurrences, and lyric texts.'

16 At least for present purposes, we distinguish between formalised musical analysis and wider, related data such as tagging. For the utility of this latter data source as a feature type, see Levy and Sandler (2007).

17 Note that we speak only of 'representation' here: there is clearly more to both chords and keys that these simple pitch class profiles.

18 The terms and abbreviations are somewhat complicated. For an introduction, see the 'Function and Transformations' section of the 'mediants' chapter of the Open Music Theory textbook.

19 For example, while RN-analysis is uniformly preferred by Anglo-American music-theory schools, in Germany, *Funktionstheorie* is more common. Engaging this demographic (which is by no means limited to students), requires support for function notation.

20 Note that the binary implementation of simple chords are handled separately in terms of one-hot encoding by the *chord quality*.

21 See de Clercq and Temperley (2011)'s corpus and analysis, especially Table 3.

22 Note also that there are many ways of expressing chord progressions. Other corners of WiR's code base include an implementation for capturing many ways of these possibilities, including expressing:

- the chords either by root, quality, or full numeral;
- the interval between those chords either by root or bass.

This includes expressing the two chords separately, for instance, seeking progressions from the tonic *specifically* to *any* chord with a root one tone higher.

23 Again, see Gotham et al. (2021b) for an assessment of how often these match in the context of different repertoires, data types and more.

ACKNOWLEDGEMENTS AND AUTHOR CONTRIBUTION

First author MG created and maintains the WiR repository with contributions from the other authors as documented by the full commit history which is publicly available on the repo here. Author MG also wrote the paper with comments from the other authors as well as editors and anonymous reviews for TISMIR (thanks to all!).

Many people have contributed to the analyses represented in WiR, and many of them chose to remain anonymous. Thanks to all contributors, and especially to Betsy Marvin, Sarah Marlow, and others at the Eastman school, with whom we have organised annual presentations and discussions about the utility of these resources for music theory classrooms, and strategies for expanding representation. We gratefully acknowledge support for this specific effort from the CNY humanities corridor every year since 2020 when author MG was based at Cornell.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Mark Gotham orcid.org/0000-0003-0722-3074
Durham University, UK

Gianluca Micchi orcid.org/0000-0002-3826-4744
Universal Music Group, NL

Néstor Nápoles López orcid.org/0000-0001-7347-2613
Sibelius/Avid & McGill University, CA

Malcolm Sailor orcid.org/0009-0008-6924-8895
Yale University, USA

REFERENCES

Bigo, L., Feisthauer, L., Giraud, M., and **Levé, F.** (2018). Relevance of musical features for cadence detection. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 355–361, Paris, France.

Chen, T.-P. and **Su, L.** (2018). Functional harmony recognition of symbolic music data with multi-task recurrent neural networks. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 90–97, Paris, France.

Chen, T.-P. and **Su, L.** (2021). Attend to chords: Improving harmonic analysis of symbolic music using Transformer-based models. *Transactions of the International Society for Music Information Retrieval*, 4(1):1–13. DOI: https://doi.org/10.5334/tismir.65

Cuthbert, M. S. and **Ariza, C.** (2010). Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 637–642, Utrecht, Netherlands.

de Berardinis, J., Meroño-Peñuela, A., Poltronieri, A., and **Presutti, V.** (2023). ChoCo: A chord corpus and a data transformation workflow for musical harmony knowledge graphs. *Scientific Data*, 10:641. DOI: https://doi.org/10.1038/s41597-023-02410-w

de Clercq, T. and **Temperley, D.** (2011). A corpus analysis of rock harmony. *Popular Music*, 30(1):47–70. DOI: https://doi.org/10.1017/S026114301000067X

Devaney, J., Arthur, C., Condit-Schultz, N., and **Nisula, K.** (2015). Theme and variation encodings with Roman numerals (TAVERN): A new data set for symbolic music analysis. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 728–734, Malaga, Spain.

Dijkstra, E. W. (1982). *Selected Writings on Computing: A Personal Perspective*. Springer-Verlag, New York. DOI: https://doi.org/10.1007/978-1-4612-5695-3

Giraud, M., Groult, R., and **Leguy, E.** (2018). Dezrann, a web framework to share music analysis. In Bhagwati, S. and Bresson, J., editors, *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'18)*, pages 104–110, Montreal, Canada.

Gotham, M., Gullings, K., Hamm, C., Hughes, B., Jarvis, B., Lavengood, M., and **Peterson, J.** (2021a). *Open Music Theory*. VIVA Pressbooks, 2nd edition.

Gotham, M. and **Jonas, P.** (2021). The OpenScore Lieder Corpus. In *Music Encoding Conference* (MEC '21).

Gotham, M. R. H. (2021). Connecting the dots: Recognizing and implementing more kinds of "Open Science" to connect musicians and musicologists. *Empirical Musicology Review*, 16. DOI: https://doi.org/10.18061/emr.v16i1.7644

Gotham, M. R. H. (2023). Chromatic chords in theory and practice. In *Proceedings of the 24th International Society for Music Information Retrieval Conference*, (forthcoming).

Gotham, M. R. H., Hentschel, J., Couturier, L., Dykeaylen, N., Rohrmeier, M., and **Giraud, M.** (2023). The 'Measure Map': an inter-operable standard for aligning symbolic music. In *Proceedings of the 10th International Conference on Digital Libraries for Musicology (DLfM '23)*, forthcoming, New York, NY, USA. ACM. DOI: https://doi.org/10.1145/3625135.3625136

Gotham, M. R. H., Kleinertz, R., Weiss, C., Muller, M., and **Klauk, S.** (2021b). What if the 'when' implies the 'what'?: Human harmonic analysis datasets clarify the relative role of the separate steps in automatic tonal analysis. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 229–236.

Harte, C., Sandler, M. B., Abdallah, S. A., and **Gómez, E.** (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pages 66–71, London, UK.

Hentschel, J., Neuwirth, M., and **Rohrmeier, M.** (2021). The Annotated Mozart Sonatas: Score, harmony, and cadence. *Transactions of the International Society for Music Information Retrieval*, 4(1):67–80. DOI: https://doi.org/10.5334/tismir.63

Huron, D. B. (2016). *Voice Leading: The Science behind a Musical Art*. MIT Press, Cambridge, Massachusetts. DOI: https://doi.org/10.7551/mitpress/9780262034852.001.0001

Levy, M. and **Sandler, M. B.** (2007). A semantic space for music derived from social tags. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 411–416, Vienna, Austria.

Lewandowski, S. (2010). ›Fallende Quintanstiege‹. *Zeitschrift der Gesellschaft fur Musiktheorie*, 7(1):85–97. DOI: https://doi.org/10.31751/508

McKay, C., Cumming, J., and **Fujinaga, I.** (2018). JSYMBOLIC 2.2: Extracting features from symbolic music for use in musicological and MIR research. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 348–354, Paris, France.

Micchi, G., Gotham, M., and **Giraud, M.** (2020). Not all roads lead to Rome: Pitch representation and model architecture for automatic harmonic analysis. *Transactions of the International Society for Music Information Retrieval*, 3(1):42–54. DOI: https://doi.org/10.5334/tismir.45

Nápoles López, N. (2017). Joseph Haydn - String Quartets Op.20 - Harmonic Analysis Annotations Dataset. https://zenodo.org/records/1095630.

Nápoles López, N., Feisthauer, L., Levé, F., and **Fujinaga, I.** (2020). On local keys, modulations, and tonicizations: A dataset and methodology for evaluating changes of key. In *7th International Conference on Digital Libraries for Musicology*, DLfM 2020, pages 18–26, New York, NY, USA. ACM. DOI: https://doi.org/10.1145/3424911.3425515

Nápoles López, N., Gotham, M. R. H., and **Fujinaga, I.** (2021). AugmentedNet: A Roman numeral analysis network with synthetic training examples and additional tonal tasks. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 404–411.

Neuwirth, M., Harasim, D., Moss, F. C., and **Rohrmeier, M.** (2018). The Annotated Beethoven Corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets. *Frontiers in Digital Humanities*, 5(16). DOI: https://doi.org/10.3389/fdigh.2018.00016

Sears, D. R. W., Verbeten J. E., and **Percival, H. M.** (2023). Does order matter? Harmonic priming effects for scrambled tonal chord sequences. *Journal of Experimental Psychology: Human Perception and Performance*. DOI: https://doi.org/10.1037/xhp0001103

Tymoczko, D., Gotham, M., Cuthbert, M. S., and **Ariza, C.** (2019). The RomanText format: A flexible and standard method for representing Roman numeral analyses. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 123–129, Delft, Netherlands.

Vatolkin, I. and **McKay, C.** (2022). Multi-objective investigation of six feature source types for multimodal music classification. *Transactions of the International Society for Music Information Retrieval*, 5:1–19. DOI: https://doi.org/10.5334/tismir.67

White, C. W. and **Quinn, I.** (2016). The Yale-Classical Archives Corpus. *Empirical Musicology Review*, 11(1). DOI: https://doi.org/10.18061/emr.v11i1.4958

TO CITE THIS ARTICLE:
Gotham, M., Micchi, G., López, N. N., and Sailor, M. (2023). When in Rome: A Meta-corpus of Functional Harmony. *Transactions of the International Society for Music Information Retrieval*, 6(1), 150–166. DOI: https://doi.org/10.5334/tismir.165

Submitted: 10 March 2023 **Accepted:** 31 July 2023 **Published:** 30 November 2023