OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration

Anonymous submission
Affiliation1
author1@uc.cl

Anonymous submission
Affiliation2
author2@uc.cl

Anonymous submission
Affiliation3
author3@uc.cl

Anonymous submission
Affiliation4
author4@uc.cl

Anonymous submission
Affiliation5
author5@uc.cl

Anonymous submission
Affiliation6
author6@uc.cl

ABSTRACT

This paper introduces OrchideaSOL, a free dataset of samples of extended instrumental playing techniques, designed to be used as default dataset for the Orchidea *framework for target-based computer-aided orchestration.*

OrchideaSOL is a reduced and modified subset of Studio On Line*, or SOL for short, a dataset developed at Ircam between 1996 and 1998. We motivate the reasons behind OrchideaSOL and describe the differences between the original SOL and our dataset. We will also show the work done in improving the dynamic ranges of orchestral families and other aspects of the data.*

1. INTRODUCTION

Target-based computer-aided orchestration is a set of techniques that help composers find combinations of orchestral sounds matching a given sound [1]. Typically, computer-aided orchestration systems consist of algorithms that compute a large number of combinations of audio samples in a dataset, corresponding to instrumental notes, trying to find the one combination that is "more similar" to the target (with respect to some metrics). Solutions to this problem are proposed as orchestral scores, often ranked by similarity between the target sound and the mixture of audio samples in the dataset. Figure 1 shows a typical workflow of a computer-aided orchestration system.

In such a context, the quality of the proposed orchestrations heavily depends on the quality and on the overall consistency of the sounds in the dataset. An essential aspect of this consistency is given by the dynamic ranges across instrumental families. Suppose, for example, that a flute note that is tagged as ***pp*** has a waveform amplitude greater than a trumpet note tagged as ***ff***. In this case, the orchestration solutions would work poorly in real-life orchestral scenarios.

Among the state-of-the-art systems for computer-aided orchestration there is the *Orch** family: *Orchidée* (2008) [2], *Orchids* (2013) [3], *Orchidea* (2017) [4], developed at Ircam (Paris) and at the Haute École de Musique (Geneva). In this paper, we focus on the last iteration, Orchidea [1] .

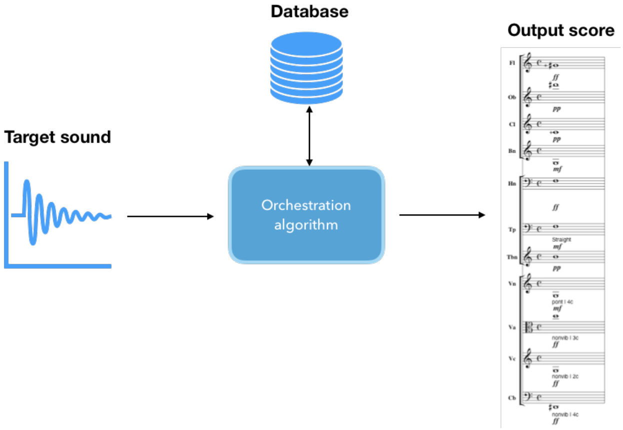


Figure 1: A diagram of a typical system for computer-aided orchestration.

Until now, Orchidea relied on the Studio On Line (SOL) dataset [5], in a specific distribution, called 0.9.2, originally created as default dataset for Orchids. This version differs from the original SOL in several regards. First, the amplitude level of each audio sample has been normalized: this means that the genuine dynamic ranges of each instrument family have been lost. Moreover, the licensing of the dataset is not free and requires a Premium account on the Ircam forum [2] .

The effects generated by the normalization of the files severely impacted the results of Orchidea. Thus, we decided to recover and process the original recordings of SOL to create three new datasets, one of them being OrchideaSOL. The three versions have different sizes and features and are distributed under different licenses:

- *TinySOL*: a small subset of SOL including only ordinario sounds, free for non-commercial usage;

- *OrchideaSOL*: the version documented in this paper, featuring a certain number of extended techniques (playing techniques), free for non commercial usages after subscription to the Ircam Forum;

[1] For more information on the Orchidea system for computer-aided orchestration, please visit: http://www.orch-idea.org

[2] See https://forum.ircam.fr/ for more information.

- *FullSOL*: the full set of samples originally present in SOL, requiring a Premium account on the Ircam forum to be used.

Although in this paper we will specifically focus on OrchideaSOL, all three vesions have been processed in the same way, in order to correct and improve the dynamic ranges and other minor problems.

2. STUDIO ON LINE: A BRIEF HISTORY

Studio On Line [5] is a dataset of instrumental sounds featuring a rich set of extended techniques commonly found throughout 20th century Western music. SOL was recorded in the *Espace de projection* of Ircam between 1996 and 1998 [6]. Over the past two decades, SOL has served as a reference dataset for many Ircam projects. The project also included software for sound processing and transformation. The head of the project was Guillaume Ballet, while the artistic managers were Joshua Fineberg (in 1997) and Fabien Lévy (in 1998).

The recordings were carried out in two phases. The first phase included the recordings of Flute, Oboe, Clarinet, Bassoon, Horn, Trumpet, Trombone, Violin, Viola, Cello and Double Bass. The second phase was planned to include more instruments and doublings [7]; yet, only Tuba, Harp, Guitar, Alto Sax and Accordion were eventually accomplished.

All instruments were recorded in a six-channel format (see Fig. 2):

- a stereo couple (track 1 and 2) was used as a reference signal;
- a proximity microphone (track 3), with minimal reverberation, was used to record the signal at a fairly high level even for very soft sounds;
- a so-called ‘internal’ microphone (track 4), either an aerial microphone placed inside the instrument or a contact microphone, was used to record sounds mostly for the purpose of acoustics studies;
- two bi-directional microphones with a figure-8 pattern (tracks 5 and 6), placed far from the musician, were used to capture the reverberation.

More than 125k samples (sampled at 48 kHz with resolution of 24 bit) were recorded [6]: ordinario sounds were sampled at least at three levels of dynamics (usually ***pp***, ***mf*** and ***ff***) and with a semitonal resolution. Woodwind instruments also included quarter tones. Most of the other playing techniques were sampled with a coarser resolution in pitch and/or in dynamics.

3. ORCHIDEASOL

We have built OrchideaSOL starting from the original version of SOL, properly cleaned and trimmed, resampled at 44.1kHz (with a bit depth of 24 bits), and made monophonic by only keeping the third channel (proximity microphone). The rationale behind this choice was to avoid the normalization introduced in later versions of the base, such as the one used by Orchids, as well as some resampling issues. The proximity microphone provided the cleanest audio recording quality. We have subsequently performed the operation described in the following subsections.

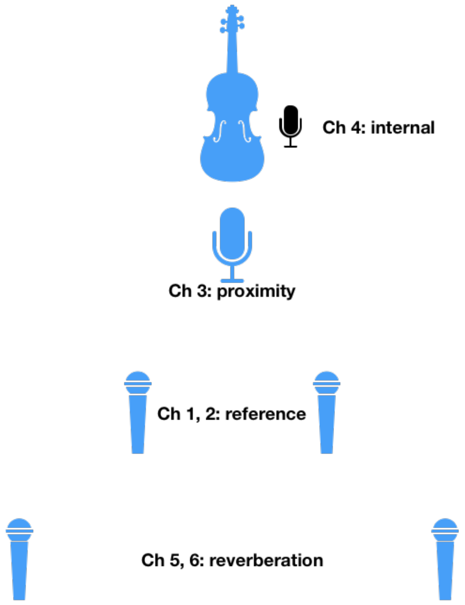


Figure 2: Spatial arrangement of the microphones during the recording sessions of SOL.

3.1 Selection

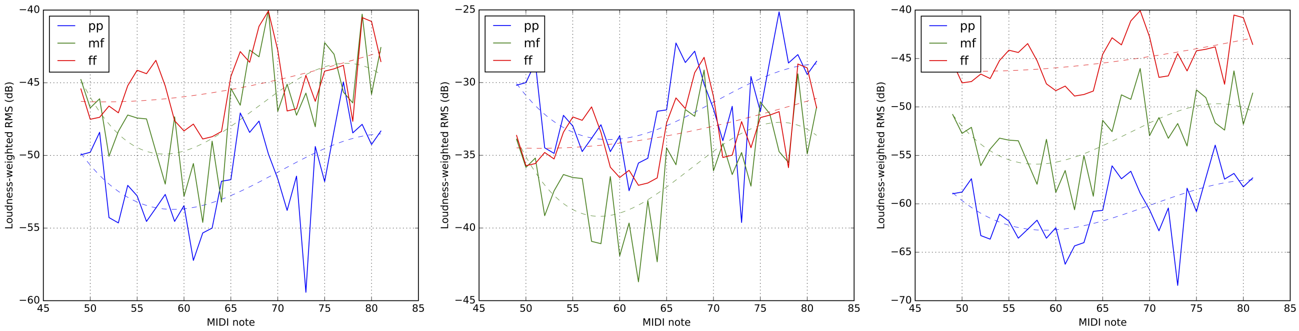
We include 12253 sound files in the free OrchideaSOL dataset, amounting to 35% of the total number of playing techniques within SOL. The choice was motivated by the desire to provide an initial set of techniques rich enough to allow experimentation with the Orchidea tools. This initial set includes: ordinario, sforzati, artificial harmonics, pizzicati, vibrati, slaps, aeolian sounds, brassy sounds, stopped sounds, flatterzunge, discolored fingerings, harmonic fingerings, notes played col legno (tratto or battuto), tremoli and bisbigliando, pedal tones, whistle tones, key clicks, jet whistles, and a few other extended techniques.

Due to the fact that currently Orchidea works with static targets, we exclude all techniques with time-varying dynamics, such as crescendi or decrescendi.

3.2 Volume compensation

Although the third channel is by far the best channel to work with, as far as recording quality, it has the important drawback that its level and positioning had been often modified during the course of recording sessions.

A subset of recordings (Alto Sax, Accordion, Harp, Guitar and Bass Tuba) had precise reports for such modifications. Most of the other instruments, unfortunately, did not. In order to retrieve more natural relationships between the instrument dynamics, whenever we could, we reverted the volume modifications according to the reports; when we could not do that, we extrapolated the rationale behind them and we tried to infer volume adjustments family-wise, with the help of analysis of volume curves across



(a)

(b)

(c)

Figure 3: RMS volume curves of the Alto Saxophone ordinario notes at three different level of dynamics, in three different SOL-related datasets: (a) the original SOL; (b) the Orchids version of SOL; (c) our OrchideaSOL. The volume is calculated by windowing the signal and weighting each window's RMS according to a measure of loudness obtained via the *pyloudnorm* library [8]. Cubic least-square regressions are shown with dashed lines.

the different levels of dynamics. We initially tried to infer the loudness differences by comparing the signal of the third microphone with the standard stereo couple, but this proved to be more intricate, invasive and error-prone than an a-posteriori analysis of volume curves.

We decided to avoid any nonlinear signal processing, such as limiters or compressors. Hence, we had to apply a global negative makeup on the dataset to avoid clipping; we recalibrated local make-up factors in order to account for macroscopic differences between families of instruments. For example, in the original dataset, a flute ***ff*** was much louder than a trumpet ***ff***. Although we are aware that our choices are far from perfection — whatever "perfection" may mean in a similar recovery task —, we believe that the volume-compensated dataset is more faithful to the relationships between instrumental dynamics than the original one. As such, we hope that it will prove more effective in orchestration tasks.

Figure 3 illustrates the application of our volume compensation procedure in the case of the Alto Saxophone. In the online repository of Orchidea, we provide a list of all the gain transformation which we applied.

3.3 Renaming

The naming in SOL was at times inconsistent and hard to parse properly. Filenames usually included four fields (instrument, playing technique, pitch, dynamics), and possibly a fifth one (other specifications), all separated by dashes. However, dashes were often used inside playing techniques themselves, as well as a description of pitch combinations, such as multiphonics of glissandi. Moreover, in few instances, some of the fields were dropped as they were not applicable to the sample at hand. In the release of Orchida-SOL, we devised a semi-automatic script to fix these issues and we performed the renaming of the sound files. [3] The general filename has been brought to the standard form:

```
<instr>-<ps>-<pitch>-<dyn>-<other>.wav
```

where `<instr>` is the abbreviated instrument name, `<ps>` is the playing technique, `<pitch>` is the pitch in textual form (under the convention that middle C is C4), `<dyn>` is the dynamics and `<other>` is any other meaningful specification (e.g. string number, alternative version number, etc.). Any among `<pitch>`, `<dyn>` and `<other>` can be replaced by the letter 'N' when such property is not applicable to the recorded file (e.g. pitch is irrelevant when playing a string instrument on the tuning pegs, and so on). All of the five property always appear in the file names (even though most of the time the `<other>` property is just 'N'). If a mute is applied, the instrument name becomes `<instr>+<mute>`.

We rewrote the names of playing techniques so as to avoid any dash or non-ASCII character. Furthermore, we harmonized discrepancies between some pairs of synonym categories, such as *sforzato* vs. *sforzando*.

3.4 Resampling

In SOL, some of the playing techniques were sampled by whole tones, minor thirds or major thirds. Moreover, some of the samples were missing altogether. Because Orchidea does not perform automatic pitch shifting of samples during the analysis process, we decided, for most of the playing techniques, to fill in the remaining notes by resampling the nearest ones, up to a tone upwards or downwards. In OrchideaSOL, we denote these resampled files by means of a trailing 'R' in their `<other>` field.

3.5 Manual corrections

Finally, whenever we came across some file which was wrongly tagged or had evident issues in volume, we applied a manual correction. The list of all the manual correction is provided in the Orchidea repository (see section 5). Some issues were found in certain folders which could not be solved (e.g. missing or corrupted samples); a list of open issues is also provided within the repository.

4. BASELINE CLASSIFICATION

Together with the dataset, we decided to include some baseline classification results on three specific tasks: instru-

[3] A renaming, slightly different from ours, was already accomplished in later version of SOL, such as the one used in Orchids. However, these versions had the important drawback of having been normalized, with seemingly no trace of the original levels.

	Instrument	Playing technique	Note
kNN	.85	.50	.80
LogReg	.85	.58	.87
SVC	.92	.73	.87
RF10	.95	.90	.90

Table 1: Classification results (accuracy) for instrument recognition, playing technique recognition and note recognition

ment recognition, playing technique recognition and note recognition. The tasks have different degrees of difficulty, given the unbalanced number of examples per class and the intrisic nature of each problem (instrument recognition has 32 classes, playing technique recognition has 89 classes and note recognition has 145 classes[4]).

The classification pipeline has been the same for each task and included, after a standardization phase, several classifiers: K-nearest neighbors (kNN), logistic regression (LogReg), support vector machines (SVC) and random forest with 10 estimators (RF10). The sounds have been analysed using 20 MFFCs; the Python code used for the classification is included in the dataset distribution and can be referred for a detailed description of the parameters of each classifier. Table 1 details the results of our experiments (accuracy). In all cases, the random forest obtained the best results. The playing technique recognition task appears to be the most difficult and the accuracies drop consequently.

It is important to remark that, in this context, we did not want to provide state-of-the-art classification results but we only wanted to give reference baselines for further experimentation. We refer to [9] on the potential of SOL and its derivatives for scientific research in machine listening.

5. DISTRIBUTION

OrchideaSOL is distributed via the Ircam Forum[5] and can be freely downloaded upon subscription.

5.1 Distribution of OrchideaSOL metadata on Zenodo

Besides its usage in musical creation, OrchideaSOL has the potential of advancing knowledge in scientific research. Indeed, the wealth of playing techniques that is afforded by OrchideaSOL, as well as the consistency of its recording conditions, makes it an ideal test bed for timbre modeling. In particular, samples in OrchideaSOL from different playing techniques are aligned in terms of onset time, fundamental frequency, and loudness level. Thus, OrchideaSOL allows to devise systematic protocols on music perception and cognition in human subjects. Furthermore, OrchideaSOL may be employed to train and evaluate machine listening software on various music information retrieval (MIR) tasks, including instrument classification, playing technique classification, and fundamental frequency estimation.

To facilitate the adoption of OrchideaSOL by the research community, we generate a metadata file in CSV format which summarizes the attributes of every audio sample. This metadata file expedites the need for writing a custom parser on the part of the MIR practitioner[6].

For the sake of research reproducibility, we provide an official split of OrchideaSOL into five non-overlapping folds, as an additional column to the metadata spreadsheet. Here, it is crucial that all folds have the same distribution of labels. To achieve this goal, we apply an algorithm named Entrofy [10], originally designed for cohort selection among human subjects. After convergence, we verify that all folds fare equally in terms of instruments, playing techniques, pitch range, and intensity dynamics.

5.2 Distribution of TinySOL in the mirdata package

Furthermore, we provide a Python module allowing every user of TinySOL to guarantee that they have access to the dataset in its pristine form. This module is integrated into the *mirdata* package [11], an open-source initiative for the reproducible usage of datasets[7]. The key idea behind *mirdata* is for the dataset curators to upload a dataset "index", in JSON format, which summarizes the list of files as well as their MD5 checksums.

To this end, we upload a copy of TinySOL to the Zenodo repository of open-access data[8]. Because Zenodo is developed by the European OpenAIRE program and operated by CERN, it has a anticipated lifespan of multiple decades. In addition, the presence of TinySOL as an unalterable dataset on Zenodo associates it to a digital object identifier (DOI), which is directly citable in scientific publications. For this reason, our implementation of the TinySOL module for *mirdata* points to the Zenodo repository as a reliable source.

A track from TinySOL may be loaded as follows:

```
from mirdata import tinysol
data_home = "mir_datasets/TinySOL"
tinysol.download(data_home=data_home)
dataset = tinysol.load()
track = dataset["Fl-ord-C4-mf-N-T14d"]
```

The corresponding waveform can be loaded (via librosa [12]) by accessing the `track.audio` property. Likewise, metadata for this track corresponds to other properties such as `instrument_abbr`, `technique_abbr`, `dynamics`, and `pitch_id`.

Because *mirdata* is version-controlled and released under a free license, it protects TinySOL against the eventuality of slight alteration, either accidental or deliberate. Indeed, the function `mirdata.tinysol.validate()` compares all MD5 checksums in the local repository against the public checksums. In case of mismatch, *mirdata* lists all files which are missing, spurious, or corrupted. Therefore, this function offers a guarantee to researchers that they are using the pristine version of TinySOL.

5.3 Distribution of pre-computed features

In addition to raw audio samples, the OrchideaSOL repository includes:

[4] This is due to the presence in the dataset of multi-pitch techniques, such as play-and-sing (playing a given note and singing another one).

[5] https://forum.ircam.fr/

[6] The metadata of OrchideaSOL can be downloaded at: https://doi.org/10.5281/zenodo.3686251

[7] The mirdata Python package can be installed with the command: `pip install mirdata`.

[8] The audio and metadata of FullSOL can be downloaded at: https://doi.org/10.5281/zenodo.3632192

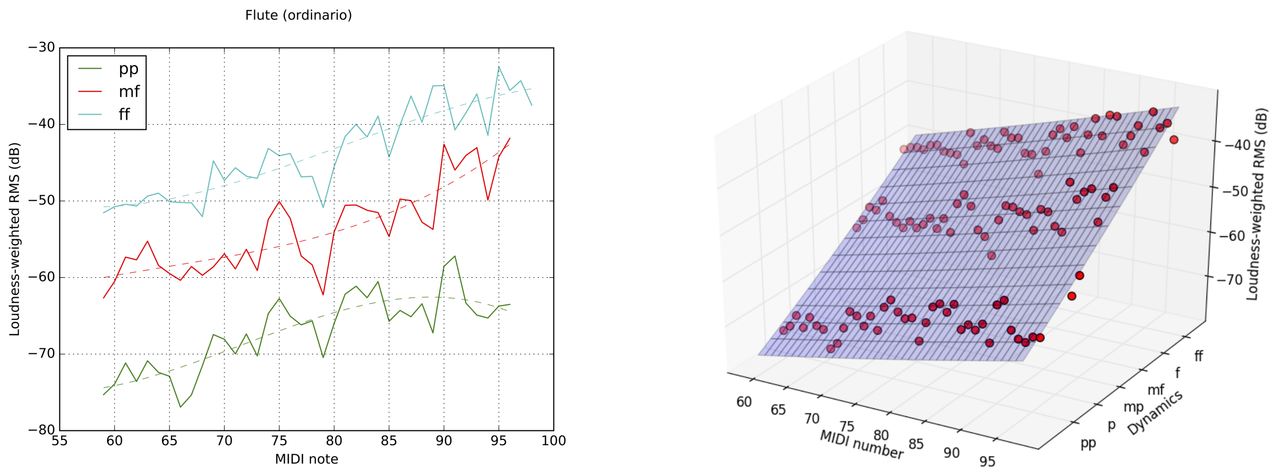


Figure 4: Volume curves for the flute (left, cubic regression lines are dashed) and three-dimensional representation of the corresponding regression surface (right).

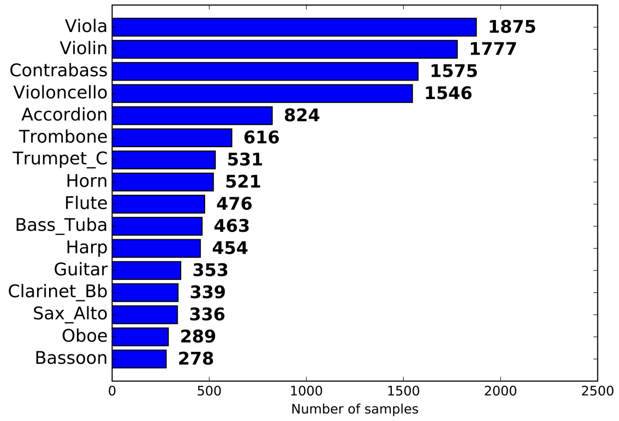


Figure 5: Distributions of samples by instrument in the OrchideaSOL dataset (samples with mutes are included in the count)

- Statistics on the number of samples (also see Figures 5 and 6).

- Volume curves for each instrument and playing techniques, with plotted cubic least-squares regression. For all the “ordinario” samples, we also provide the coefficient of the quadratic two-dimensional regression formula for the loudness-weighted RMS (L_{dB}):

$$L_{\text{dB}} = \alpha + \beta M + \gamma D + \delta M D + \phi M^2 + \psi D^2$$

where M is the MIDI number and D is a numeric value assigned to each dynamic marking according to the convention

$$\ldots,\quad \boldsymbol{pp} = -2.5,\quad \boldsymbol{p} = -1.5,\quad \boldsymbol{mp} = 0.5,\quad \boldsymbol{mf} = 0.5,\quad \boldsymbol{f} = 1.5,\quad \boldsymbol{ff} = 2.5,\quad \ldots$$

Figure 4 show some examples of volume curves for the flute (ordinario). The quadratic regression for its loudness-weighted RMS is

$$L_{\text{dB,flute}} \approx -82.3301 + 0.2732M + 3.0831D + \\ + 0.0194MD + 0.0008M^2 + 0.0823D^2$$

These formulas can be also used to roughly model instrumental dynamics in other contexts, such as synthesis or sampling.

- Spectral analysis: for each sample, we provide the first 1024 amplitude bins of its average spectrum (analyzed with window size: 4096 samples, hop size: 2048 samples);

- MFCC analysis: for each sample, we provide the average of the first 20 MFCCs (analyzed with window size: 4096 samples, hop size: 2048 samples).

- Spectral envelope: for each sample, we provide the first 1024 bins of the average spectral envelope (analyzed with window size: 4096 samples, hop size: 2048 samples); the spectral envelope is computed by applying a lowpass window of 80 coefficients (liftering) to the cepstrum and by taking again the Fourier transform; briefly, given the signal S:

$$E = FFT(W_{LP}(FFT^{-1}(\log(|FFT(S)|)).$$

- Spectral peaks: for each sample, we provide the average of the first 120 peaks of the magnitude spectrum analyzed with window size: equal to 4096 samples and hop size equal to 2048 samples.

- Spectral moments: for each sample, we provide the average of the first 4 spectral moments [13]: centroid, spread, skewness and kurtosis, analyzed with window size equal to 4096 samples and hop size equal to 2048 samples).

6. CONCLUSION

We have introduced OrchideaSOL, a free subset of the SOL dataset modified and tailored to be reliable in target-based

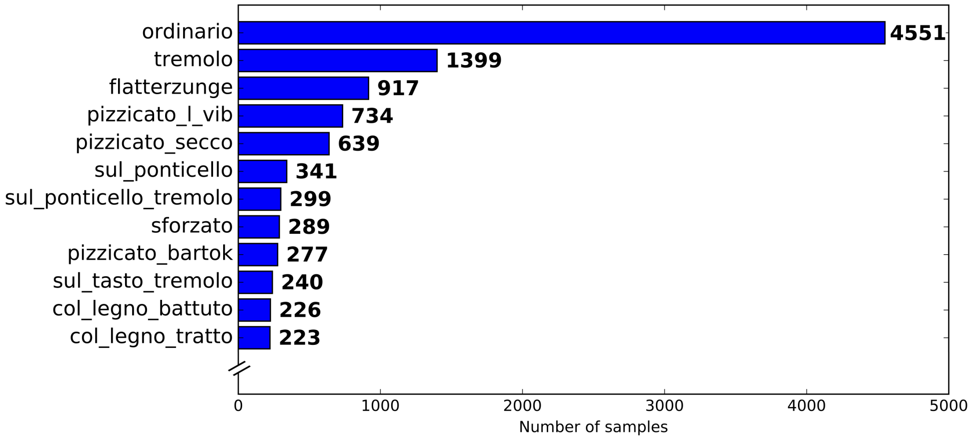


Figure 6: Distributions of samples by playing technique in the OrchideaSOL dataset. Only playing techniques with more than 200 entries are displayed (12 out of 52).

computer-aided composition tasks. We expect the generated scores to be more faithful in real-life orchestral scenarios than with the previous SOL dataset. Moreover, we believe that OrchideaSOL has a strong potential for advancing scientific research in music cognition, music information retrieval, and generative models for sound synthesis.

Acknowledgments

We would like to thank Hugues Vinet, Jean-Louis Giavitto and Gregoire Beller for sharing with us the collection of internal Ircam reports on the SOL recordings and for agreeing to release OrchideaSOL for free.

7. REFERENCES

[1] Y. Maresz, “On computer-assisted orchestration,” *Contemporary Music Review*, vol. 32, no. 1, pp. 99–109, 2013.

[2] G. Carpentier and J. Bresson, “Interacting with symbol, sound, and feature spaces in orchidée, a computer-aided orchestration environment,” *Computer Music Journal*, vol. 34, no. 1, pp. 10–27, 2010.

[3] M. Caetano and C.-E. Cella, “Imitative computer-aided musical orchestration with biologically inspired algorithms,” in *Handbook of Artificial Intelligence for Music*, E. Miranda, Ed. Springer, 2020, to appear.

[4] J. Gillick, C.-E. Cella, and D. Bamman, “Estimating Unobserved Audio Features for Target-Based Orchestration,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019.

[5] G. Ballet, R. Borghesi, P. Hoffmann, and F. Lévy, “Studio online 3.0: An internet ”killer application” for remote access to ircam sounds and processing tools,” *Journée d’Informatique Musicale (JIM)*, 1999.

[6] F. Lévy, “STUDIO ON LINE: Une présentation générale (internal report),” Ircam, Tech. Rep., 1999.

[7] J. Fineberg, “IRCAM Instrumental Data Base (internal report),” Ircam, Tech. Rep., 1996.

[8] C. Steinmetz, “csteinmetz1/pyloudnorm: 0.1.0,” Nov. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3551801

[9] V. Lostanlen, J. Andén, and M. Lagrange, “Extended playing techniques: The next milestone in musical instrument recognition,” in *Proceedings of the International Conference on Digital Libraries for Musicology (DLfM)*, 2018.

[10] D. Huppenkothen, B. McFee, and L. Norén, “Entrofy Your Cohort: A Data Science Approach to Candidate Selection,” *arXiv preprint arXiv:1905.03314*, 2019.

[11] R. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, “mirdata: Software for Reproducible Usage of Datasets,” in *Proceedings of the International Society on Music Information Retrieval (ISMIR) Conference*, 2019.

[12] B. McFee, V. Lostanlen, M. McVicar, A. Metsai, S. Balke, C. Thomé, C. Raffel, A. Malek, D. Lee, F. Zalkow, K. Lee, O. Nieto, J. Mason, D. Ellis, R. Yamamoto, S. Seyfarth, E. Battenberg, V. Morozov, R. Bittner, K. Choi, J. Moore, Z. Wei, S. Hidaka, P. Friesch, F.-R. Stöter, D. Hereñú, T. Kim, M. Vollrath, and A. Weiss, “librosa/librosa: 0.7.2,” Jan. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3606573

[13] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” *CUIDADO IST Project Report*, vol. 54, no. 0, pp. 1–25, 2004.