

ISMIR Tutorial Proposal

Chris Cannam, Luís A. Figueira, Marco Fabiani,
Mark D. Plumbley, and Simon Dixon
Centre for Digital Music, Queen Mary University of London
`{firstname.lastname}@eecs.qmul.ac.uk`

March 9, 2012

1 Title

Reusable software and reproducibility in music informatics research

2 Motivation

The need to develop and reuse software to process data is almost universal in music informatics research. Many methods, including most of those published at ISMIR, are developed in tandem with software implementations, and some of them are too complex or too fundamentally software-based to be reproduced readily from a published paper alone. For this reason, it is helpful for sustainable research to have software and data published along with papers.

In practice, non-publication of code and data is still the norm and research software is commonly lost following publication of the associated methods.

For the Sound Software project¹ we carried out a survey of UK audio and music researchers in 2011. Of those respondents who reported both developing software during research and taking steps to reproducibility for their publications, only 35% reported having in fact published any of their code. Respondents cited as obstacles to publication of code lack of time, copyright restrictions, and the potential for future commercial use. A broader study of research across several subject areas by the UK Research Information Network additionally identified lack of evidence of benefits, cultures of independence and competition, and quality concerns as inhibiting factors.

We identified a number of barriers to the publication of software and data, including:

- lack of education in software development and consequently of confidence that code is of publishable quality;
- lack of facilities and tools to enable collaborative development and to support a familiarity with sharing and publishing code;
- lack of incentive to distribute software and data;
- reusability problems caused by platform incompatibilities.

During this tutorial we will discuss these problems, outline a possible course of action that researchers and research groups can take in order to mitigate each of these barriers, and present a practical, hands-on session in which attendees

¹<http://soundsoftware.ac.uk/>

can familiarise themselves with some of the tools and methods involved and gain confidence with using them for their own work.

3 Outline of the tutorial content

This tutorial will be in three parts:

- An **introduction and overview** discussing the motivation for reusable software and data in research and providing an overview of some methods, tools and facilities available to researchers for this purpose;
- A **hands-on session** in which attendees are encouraged to try out some of these methods in code;
- A review and discussion of **practical issues** in ensuring that publication of data and code actually occurs, relevant also to research group leaders.

3.1 Introduction and overview

In the first part of the tutorial, we will first set out motivations for publishing software and code, and then discuss problems faced by researchers in trying to do so, taking into account their consequences for scientific rigour.

We then give an overview of software tools, facilities and methods available for researchers to assist with collaborative development and software publication, including:

- Version control software: The concepts; practical advantages; overview of Mercurial, Git, Subversion; hosting facilities such as Github, Bitbucket, or (for UK researchers) our own `code.soundsoftware.ac.uk`;
- Unit testing and managing provenance and reproducibility for code;
- Data management: principles, repositories, and versioning;
- Software and data licences: commonly-used open-source licences; pros and cons of GPL and BSD licensing schemes; Creative Commons

3.2 Hands-on session

The second part of the tutorial will be a hands-on session in which attendees will get the opportunity to work through an example with real code.

A “toy” music informatics programming problem will be presented, with sample code and data available, and attendees will pair up to:

- Implement it in Python or MATLAB/Octave (according to their preference) using a very simple unit testing regime;
- Place the code under version control using a local repository in a distributed version control system;
- Tag the code and make a record associating the software version with its output data version;
- Tweak the algorithm and record the updated software and data versions accordingly;
- Place the resulting software under a standard open-source software licence;
- Follow a simple “release procedure” to produce a code and data release.

3.3 Practical issues

The third part of the tutorial will open out the discussion into the wider field of reproducible publication, and into areas of policy and actions that research groups and research leaders may wish to consider.

This section will therefore cover:

- Publication mechanisms for reproducible research:
 - Open-access journal papers;
 - Self-archiving;
 - Technical reports;
 - Copyright issues relating to journal or book publication;
 - Mechanisms for associating software with the paper publication;
 - Identifying specific versions of software or data with a publication.
- Publication policies for research group leaders:
 - Why publish software and data?;
 - What software and data should be published, and when?
 - Institutional assistance with publication barriers;
 - The research community.

4 Intended and expected audience

The primary audience for this tutorial is researchers within the music informatics community who develop or reuse software and data during their day-to-day research.

We believe that an overwhelming majority of material submitted to ISMIR requires software to be developed during research. Given results showing that most researchers are self-taught in software development, and in light of the reasons researchers report as to why they do not publish software, we think that a large proportion of the active researchers represented at ISMIR regardless of subject focus will find the material in our tutorial of interest.

Our tutorial is also highly relevant to research supervisors and research group leaders, because of its implications in terms of both institutional and group policy and guidance for research students.

5 Short biography of the presenters

5.1 Experience in this area

The presenters manage the Sound Software project and Sustainable Management of Digital Music Research Data project² in the Centre for Digital Music (C4DM) at Queen Mary University of London.

The Sound Software project is an EPSRC-funded initiative to assist researchers to manage software code in a more sustainable manner, based at the C4DM but with the whole UK audio and music research community as its focus.

The Sustainable Management of Digital Music Research Data project is a JISC-funded pilot data-management project focusing on data published by the C4DM.

²<http://rdm.c4dm.eecs.qmul.ac.uk/>

The presenters have extensive experience in audio and music research and software development, and have given workshops on sustainable software development in research at the C4DM and elsewhere in the UK.

5.2 The presenters

Chris Cannam is the principal developer for the Sound Software project and code hosting site.³ He is a software developer with many years of commercial and open-source cross-platform development experience. While at C4DM he has developed software including the widely-used Sonic Visualiser audio analysis and visualisation application.

Luís Figueira is a software developer with several years of experience with C/C++, Ruby on Rails, Scheme, Web technologies and databases. He has an MSc in Electrotechnical and Computer Engineering from Instituto Superior Técnico in Lisbon, where he specialized in digital signal processing with a focus on speech synthesis.

Dr Marco Fabiani is a post-doctoral Research Assistant at C4DM working on the Sustainable Management of Digital Music Research Data project. He recently completed his PhD in Computer Science - Speech and Music Communication (KTH, Stockholm) with a thesis on interactive computer-based music performance, and has worked on topics including audio signal processing, music information retrieval, HCI, and sound perception.

Prof Mark Plumbley is Director of C4DM and leads the Sound Software initiative. His work in audio signal analysis includes beat tracking, music transcription, source separation and object coding, using techniques such as neural networks, independent component analysis, sparse representations and Bayesian modelling. He is Chair of the International Independent Component Analysis Steering Committee, a member of the IEEE Machine Learning in Signal Processing Technical Committee, and an Associate Editor for IEEE Transactions on Neural Networks. He leads the ICA Research Network and Digital Music Research Network.

Dr Simon Dixon leads the Music Informatics area of C4DM and the Sustainable Management of Digital Music Research Data project. His research interests cover various aspects of music informatics, including high-level music signal analysis and the representation of musical knowledge. He has been General Co-Chair of the Dagstuhl Seminar on Multimodal Music Processing and Computer Music Modeling and Retrieval, Programme Co-Chair for ISMIR 2007, and co-presenter of the ISMIR 2006 tutorial on Computational Rhythm Description.

6 Any special requirements

Attendees will be encouraged to bring and use laptops, so sufficient space and network capacity would be welcome.

It would be nice to separate the three parts of the tutorial with coffee and biscuit breaks!

7 Contact information

Please contact Chris Cannam, chris.cannam@eecs.qmul.ac.uk.

³<http://code.soundsoftware.ac.uk>