

# ISMIR Tutorial Proposal

Mark D. Plumbley, Chris Cannam, and Luís A. Figueira  
Centre for Digital Music, Queen Mary University of London  
`{mark.plumbley, chris.cannam, luis.figueira}@eecs.qmul.ac.uk`

March 8, 2012

## 1 Title

**Reusable software and reproducibility in music informatics research**

## 2 Outline of the tutorial content

This tutorial will be in three parts:

- An **introduction and overview** discussing the motivation for reusable software in research and providing an overview of some methods, tools and facilities available to researchers for this purpose;
- A practical **hands-on example** section in which attendees are encouraged to try out some of these methods in code;
- A **review and discussion** on the subject of publications policy, relevant also to research group leaders.

### 2.1 Introduction and overview

In the first part of the tutorial we will discuss some of the problems faced by researchers in developing and reusing software in their research, and their consequences for scientific work. We will then outline some of the options available for researchers to help overcome barriers to software reuse.

#### 2.1.1 The present situation and what we can do about it

We will talk about findings which show limited levels of collaborative development and software publication in audio and music research, and present and discuss research and survey data pointing to some common causes for the lack of publication and eventual disappearance of research software code.

For example, our survey of UK audio and music researchers in 2011 found that even among 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. Our respondents cited as obstacles to the publication of code lack of time, copyright restrictions, and the potential for future commercial use. (A broader study into science research across several subject areas by the UK Research Information Network additionally identified the lack of evidence of benefits, cultures of independence and competition, and quality concerns as inhibiting factors.)

We will identify a number of barriers to the publication of code, including the lack of education and confidence with code, lack of facilities and tools to support collaborative development, lack of incentive to distribute software, and reusability problems caused by platform incompatibilities. We will outline a possible course of action that researchers and research groups can take in order to mitigate each of these barriers, including focused small-scale training programmes, the use of version control software, and actions that create an association between published software and citeable publications.

### **2.1.2 Software, tools, and facilities**

This section will present an overview of methods, tools and facilities available to 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
- Software licences: commonly-used open-source licences; the pros and cons of GPL and BSD licensing schemes
- Data management: principles and repositories

## **2.2 Hands-on examples**

In this section, the second of the three, attendees will get the opportunity to work through an example using real code.

A toy MIR-related programming problem will be presented, and attendees will pair up to:

- Implement it in either Python or MATLAB/Octave using a very simple unit testing regime;
- Place the code under version control using a local repository in a distributed version control system;
- Make the appropriate changes to place the result under a standard open-source software licence;
- Tag the code and follow a simple "release procedure" to produce a source code package.

## 2.3 Review and discussion

Following a review of the results of the hands-on example, we 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
  - Publishing software in such a way that its relationship with the written publication is apparent
  - Associating specific versions of software or data with a publication
- Publication policies for research group leaders:
  - Why publish?
  - Institutional assistance with publication barriers
  - The research community

## 3 Intended and expected audience

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

We believe that an overwhelming majority of material submitted to ISMIR required 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.

## 4 Short biography of the presenter(s)

The presenters manage the Sound Software project (<http://soundsoftware.ac.uk/>), an initiative to assist researchers in audio and music fields in the UK to manage

software code in a more sustainable manner. Based at the Centre for Digital Music (C4DM) at Queen Mary, University of London, they have extensive experience in audio and music research (including music information retrieval) and software development, and have given workshops on sustainable software development in research at the C4DM and elsewhere in the UK.

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. Prof Plumbley 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.

Chris Cannam is a software developer with 15 years commercial and extensive open-source and cross-platform development experience. While at the C4DM he has worked on the widely-used Sonic Visualiser audio analysis and visualisation application; Sonic Annotator, a tool for batch extraction of meaningful features from audio files; the Vamp plugin API for audio feature extraction, and many plugins using this API; and tools and ontologies for music description using RDF within the Semantic Web.

Luis Figueira is a software developer with more than 5 years of experience with C/C++, Ruby on Rails, Scheme, Web technologies and databases. He has an MSc in Electrotechnical and Computers Engineering from Instituto Superior Tcnico in Lisbon, where he specialized in digital signal processing with a focus on speech synthesis. Luis has recently worked in a speech technology spin-off and an open-source web development company.

## **5 Any special requirements**

## **6 Contact information**