

Mika Kuuskankare

SDIF

PWGL SDIF Library
(v 1.11)



August 5, 2011

Contents

1	Introduction	2
1.1	Quick Reference	2
1.1.1	Box	2
1.1.2	Input Boxes ('options')	2
2	Todo	2
3	Tools	2
4	Querysdif	3
5	Sdifextract	4
6	Resonance-Models	5
7	Chord-Sequence	6
8	Sdif-Boxes	8
A	Box Reference	9

1 Introduction

SDIF is a small and simple PWGL library that provides the users with a box interface to the tools provided by the official SDIF distribution. It allows us to read in a lispy way the data contained by .sdif files. For SDIF documentation and any further information, see: <http://sdif.sourceforge.net/>

The SDIF test files (distributed inside the 'SDIF-Tests' folder) are from CNMAT: <http://archive.cnmat.berkeley.edu/SDIF/alpha/SDIF-Files.tar.gz>

The SDIF library itself is implemented using the SHELL library. See the SHELL documentation (Tutorial/Special-Boxes/Shell) for further information.

1.1 Quick Reference

1.1.1 Box

- +, to add an option.
- -, to remove THE LAST option.
- h, to read the man entry.

1.1.2 Input Boxes ('options')

- ARROW UP/DOWN, move the selected input-box ('option') forward or backward.
- DELETE, remove an input-box ('option').

2 Todo

3 Tools

The SDIF library implements two boxes: (1) `querysdif` (2) `sdifextract`
`querysdif` can be used to view a summary of the data stored in an SDIF-file.
`sdifextract`, in turn, can be used to extract the data stored in a given file.

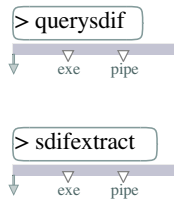


Figure 1: 09-tools

4 Querysdif

The purpose of the querysdif box is to display a summary of data in an SDIF-file. The possible options are:

- a
view ASCII chunks
- d
view data
- b
view data brief (output in SDIF selection syntax)
- t <sdif types file> specify file with additional sdif types
- h
help

The information is printed in the PWGL Output.

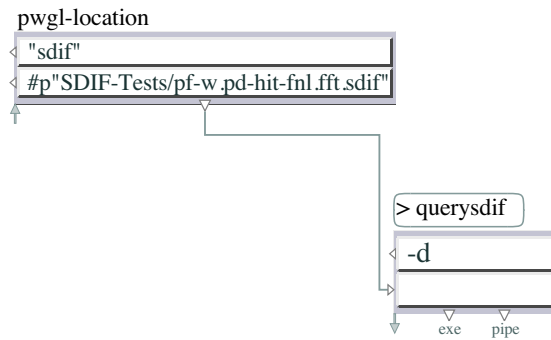


Figure 2: 10-querysdif

5 Sdifextract

In this patch we read in FFT information stored in an SDIF file. The SDIF file is given in (1). `sdifextract` box in (2) has, besides the mandatory pathname argument, three options. The `'-data'` option instructs the box to return only the data without times. The `'-t'` option gets as an argument a time range, i.e., we read in only the frames between 0 and 5 milliseconds. The `'-m'` option allows us to handle only specific types of matrices, the type `'1GB0'` in this case. As we are only interested in the data, the structured sublists are simply flattened in (4). The `'2D-constructor'` in (5) creates individual breakpoint-functions of each of the FFT-frames. The result is shown in the `'2D-Editor'` at the bottom of the patch.

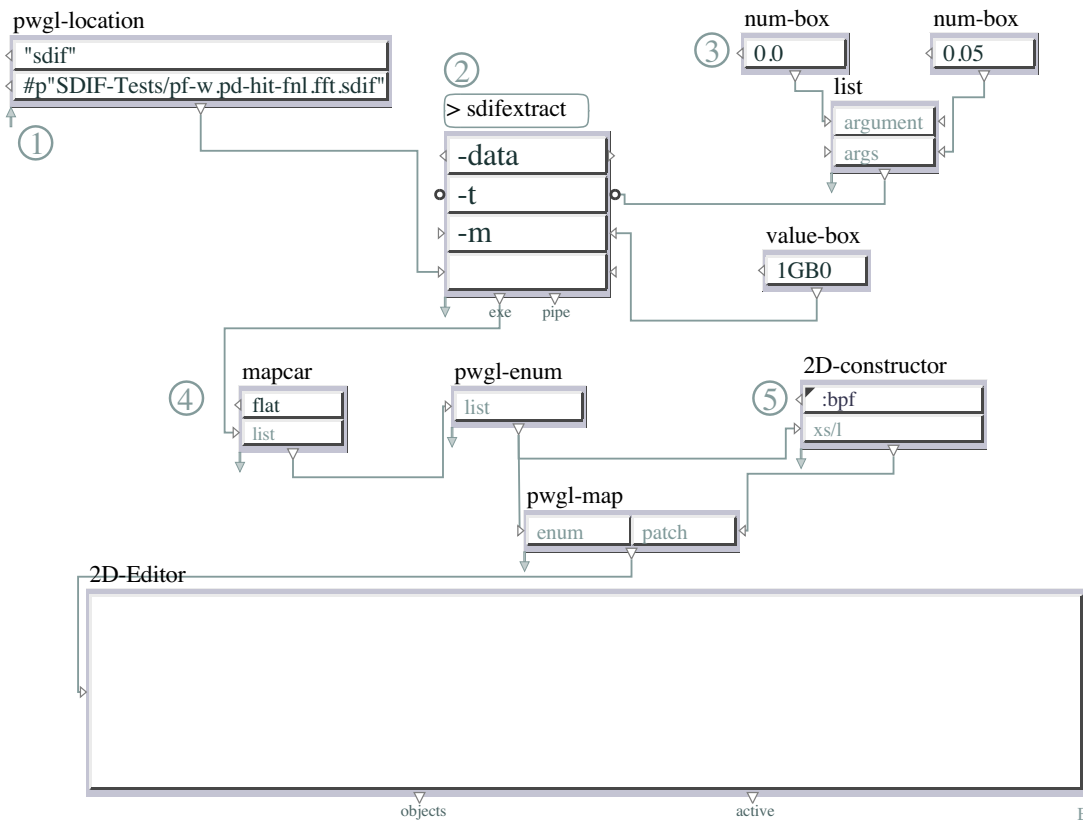


Figure 3: 11-sdifextract

6 Resonance-Models

In this patch we use the 'sdifextract' module to read analysis data that is used to control a resonator bank. The source sdif file can be chosen from the 'menu-box'.

The resonator data can be filtered as follows: (1) filter by amplitude range (2) filter by taking 'count' loudest partials (3) filter by frequency range

Choose one of these options using the master switch box 'amp/count/freq'.

In the synthesis part you can choose to excite the resonator bank with either impulse or noise. Choose one of these options using the master switch box 'imp/noise'.

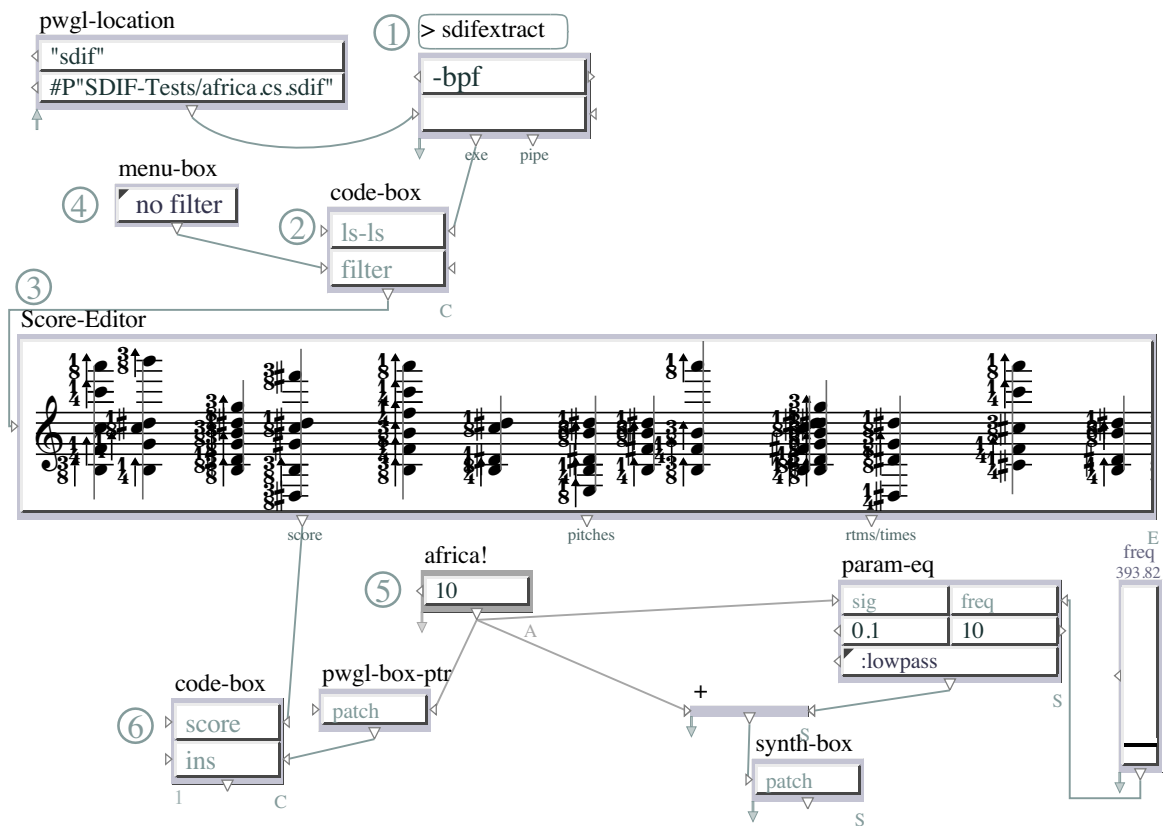


Figure 5: 13-chord-sequence

8 Sdif-Boxes

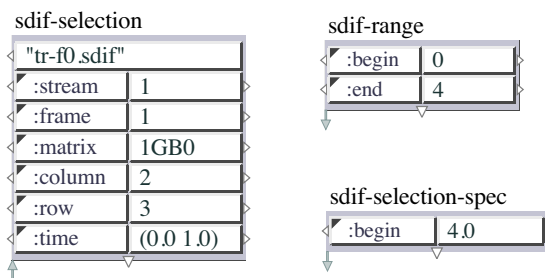


Figure 6: sdif-boxes

A Box Reference

sdif-range

arglist: (&key begin end delta)

package: SDIF

menu: SDIF

sdif-selection

arglist: (filename &key stream frame matrix column row time)

package: SDIF

menu: SDIF

sdif-selection-spec

arglist: (&key begin end list range)

package: SDIF

menu: SDIF