Documentation of Redmonster Output File

Timothy A. Hutchinson

October 24, 2014

1 Introduction

This document describes the format of the *redmonster* file, which contains the spectroscopic redshifts and classifications of the `redmonster` software.

2 File Format

2.1 File naming convention

Files from `redmonster` will generally follow the naming scheme

redmonster-pppp-mmmmm.fits

where *pppp* is the 4-digit SDSS plate number, and *mmmmm* is the 5-digit MJD. These correspond with the *spPlate* input file and and cannot be changed by the user.

In the case where a file with the above naming convention exists in the given path, the default behavior of the software is to overwrite it. However, the user may elect to leave the older file intact, in which case the new file will be written as

redmonster-pppp-mmmmm-YYYY-nn-dd_HH:MM:SS.fits

where *YYYY*, *nn*, *dd*, *HH*, *MM*, and *SS* are the year, month, day, hour, minute, and second, respectively, of the time at which the file was written.

2.2 File type

All `redmonster` outputs are uncompressed FITS files with all relevant information in the primary HDU header and the first and second BIN tables. The file size is approximately 20 Mb.

2.3 File contents

The general structure of the file is as follows:

HDU0	NULL	Empty
HDU1	Binary FITS Table	Object redshifts and classifications
HDU2	NFIBERS x NPIX float image	Best-fit template model for each object

The primary HDU header is identical to that of the *spPlate* files. The header keywords are as follows:

SIMPLE	FITS STANDARD
BITPIX	PIXEL
NAXIS	NUMBER OF AXES
EXTEND	
TAI	1st row - Number of seconds since 17 Nov 1858
RA	1st row - Right ascension of telescope boresight
DEC	1st row - Declination of telescope boresight
EQUINOX	
RADECSYS	
AZ	1st row - Azimuth of telescope

2.4 Header structure and requirements

The following primary header keywords are **required** to be present and defined as specified, in addition to header keywords required by the FITS standard itself:

`CRPIX1`: Shall be set to the value 1, referencing the first sample point ("pixel") along the wavelength axis (one-based indexing).

`CRVAL1`: Shall specify the **base-10 logarithm** of the central wavelength **in vacuum Angstroms** of first pixel along the wavelength axis.

`CDELT1`: Shall specify the pixel-to-pixel increment in **base-10 logarithm** of vacuum wavelength from one pixel to the next along the wavelength axis.

The following primary header keyword is **required** in the case that the physical normalization of the templates is meaningful:

`BUNIT`: String giving the units of the template spectra.

The following primary header keyword is **required** in the case that the templates are given with respect to air wavelengths rather than vacuum:

`AIRORVAC`: String that is either `'air'` or `'vac'` depending upon wavelength convention. (The only significant value is `'air'`, since anything else, including the absence of this keyword, will be interpreted as `'vac'`.)

The following primary header keywords are supported as optional but **recommended** in some combination as appropriate:

`CNAMEn`: For $n > 1$, a string giving the name of the physical parameter coordinate along the n^{th} axis.

`CUNITn`: For $n > 1$, a string giving the units of the physical parameter coordinate along the n^{th} axis.

`CRPIXn, CRVALn, CDELTn`: For $n > 1$, specifying the physical parameter baselines for axes corresponding to regularly gridded numerical physical parameters.

`PVn_j`: For $n > 1$ and for the case of irregularly gridded numerical physical parameters along axis n , specifies the parameter value of the j^{th} point along the n^{th} axis. The index j shall begin with 1 and increase in integer steps up to the size of the n^{th} axis, with one keyword per grid step. Note that the FITS standard currently limits the maximum size of axes that can be represented in this manner to 99, and that leading zero-padding is not allowed.

`PSn_j`: For $n > 1$ and for the case of non-numerical physical parameters along axis n , specifies the parameter string value of the j^{th} point along the n^{th} axis. The index j shall begin with 1 and increase in integer steps up to the size of the n^{th} axis, with one keyword per grid step. Note that the FITS standard currently limits the maximum size of axes that can be represented in this manner to 99, and that leading zero-padding is not allowed.

`Nn_j`: For $n > 1$ and for the case of arbitrary labels along axis n, specifies the label string for the j^{th} point along the n^{th} axis. The index j shall begin with 1 and increase in integer steps up to the size of the n^{th} axis, with one keyword per grid step. These keywords do not belong to the FITS standard, and can accommodate dimensionality up to 999.

For any given axis beyond the first (wavelength) axis, the `redmonster` code will use information from the above keywords to construct parameter-grid baselines. The precedence for establishing the baseline for each axis is first for `CRPIXn, CRVALn, CDELTn`, then for `PVn_j`, then for `PSn_j`, then for `Nn_j`, then for a one-based integer baseline in the absence of a valid set of keywords of any of the preceding specified types.

3 Implementation

Reader and writer routines for `ndArch` files conforming to this proposed standard are implemented in the `redmonster.datamgr.io` module as `read_ndArch` and `write_ndArch`. (The location of these routines is subject to change with future package reorganizations.) Detailed documentation of these file-handling routines can be found in their embedded docstrings.

A script called `test_ndArch.py` is also provided, which generates a file named `ndArch-TEST-v00.fits`, which in turn provides a unit test of the functioning of the `read_ndArch` routine.