ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-spectral-2dim-core



Peter Reutemann

November 21, 2026

©2017-2026 THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Except where otherwise noted, this work is licensed under
http://creativecommons.org/licenses/by-sa/4.0/

Contents

1 Introduction **7**

2 Flow **9**

3 Visualization **13**

4 Maintenance **15**

5 Syntax highlighting **17**

Bibliography **19**

List of Figures

3.1 Spectrum Explorer 13

Chapter 1

Introduction

The *adams-spectral-2dim-core* module offers a comprehensive suite of tools and data processing of two-dimensional spectra, such as generted from near-infrared[2], mid-infrared[3] and X-ray fluorescence[4] instruments.

Chapter 2

Flow

The following conversions are available:

- *GetSpectrumToFormat* – extracts and forwards the format of a spectrum.
- *MultiSpectrumToSpectra* – turns a single multi-spectrum into an array of regular spectra.
- *ReportToSampleData* – converts a simple report into a sample data object.
- *SampleDataArrayToMap* – converts a sample data array to a map, using the sample ID as key.
- *SetSpectrumToFormat* – updates the format of spectra passing through.
- *SpectraToMultiSpectrum* – turns an array of spectra into a single multi-spectrum.
- *SpectrumToArray* – turns a spectrum into a float array (either wave numbers or amplitudes).
- *SpectrumToJson* – turns a spectrum into a JSON object.
- *SpectrumToSpreadSheet* – turns a spectrum into a spreadsheet with two columns: wave numbers and amplitudes.
- *SpreadSheetColumnsToSampleData* – generates sample data objects from the columns in a spreadsheet.
- *SpreadSheetColumnsToSpectra* – generates spectra from the columns in a spreadsheet.
- *SpreadSheetRowsToSampleData* – generates sample data objects from the rows in a spreadsheet.
- *SpreadSheetRowsToSpectra* – generates spectra from the rows in a spreadsheet.
- *SpreadSheetToSpectrum* – generates a spectrum from a spreadsheet with two columns containing the wave numbers and amplitudes.
- *WekaPredictionContainerToEvaluationContainer* – turns the prediction container into an evaluation container.

The following sources are available:

- *InstrumentSupplier* – returns all instrument names from the database.
- *NewSpectrum* – creates an empty spectrum.
- *SpectrumIdSupplier* – outputs database or sample IDs from spectra stored in the database.

- *SpectrumSupplier* – outputs spectra stored in the database.
- *OrphanedSampleDataIdSupplier* – allows the retrieval of sample IDs of sample data that have no spectra associated.
- *SQLSpectrumIdSupplier* – outputs IDs (integer or string) generated from a SQL query.

The following transformers are available:

- *BulkSampleDataDbWriter* – writes sample data to the database like *SampleDataDbWriter*, but with much higher throughput.
- *BulkSpectrumDbWriter* – writes spectra to the database like *SpectrumDbWriter*, but with much higher throughput.
- *Cleaner* – cleans a Weka dataset with a specific cleaner and performs checks on single Instance objects.
- *DeleteDbSampleDataValue* – removes the specified fields from the sample-data table for a specific sample ID.
- *DeleteSampleDataValue* – removes the specified fields from the sample data passing through.
- *DeleteSampleDataValueByExpression* – deletes a specific field from sample data if the boolean expression evaluates to true.
- *DeleteSpectrum* – removes a spectrum, identified by database or sample ID, from the database.
- *Evaluator* – trains an evaluator on a Weka dataset and applies the evaluator on single Weka Instance objects.
- *GetSampleDataValue* – retrieves a single value from a sample data object.
- *GetSpectrumAmplitude* – retrieves the amplitude of the specified wave number.
- *InstanceGenerator* – turns a spectrum into a Weka Instance.
- *InstanceToSpectrum* – turns a Weka Instance into a spectrum.
- *MergeSampleDataFromMap* – merges the sample data from a map in storage with the spectrum/sample data object passing through.
- *MultiSpectrumAdd* – adds a spectrum obtained from storage or callable actor to the multi-spectrum passing through.
- *MultiSpectrumFilter* – applies a filter to a multi-spectrum.
- *MultiSpectrumOutlierDetector* – applies outlier detectors to multi-spectra.
- *MultiSpectrumRemove* – removes spectra from a multi-spectrum that match expressions on sample ID, type or format.
- *MultiSpectrumReportFilter* – filters the reports of a multi-spectrum.
- *PostProcessor* – allows the post-processing of Weka Instances.
- *SampleDataDbReader* – reads sample data from the database.
- *SampleDataDbWriter* – writes sample data to the database.
- *SampleDataFileReader* – loads sample data from disk.
- *SampleDataFileWriter* – writes sample data to disk.
- *SampleDataValueDbWriter* – writes specific sample data fields to the database.
- *SetSampleDataValue* – sets the value for the specified sample data field.
- *SetSpectrumAmplitude* – set the amplitude of the specified wave number or adds a new spectrum point altogether.
- *SpectrumAppend* – appens the incoming spectrum to the one in storage (latter doesn't have to be present).

- *SpectrumDbReader* – loads a spectrum from the database.
- *SpectrumDbWriter* – writes a spectrum to the database.
- *SpectrumDiff* – calculates the difference between the two spectra received as array and outputs the difference as spectrum.
- *SpectrumFeatureGenerator* – for generating features from a spectrum.
- *SpectrumFileReader* – loads a spectrum from disk.
- *SpectrumFileWriter* – writes a spectrum to disk.
- *SpectrumFilter* – applies a filter to the spectrum passing through.
- *SpectrumInfo* – outputs information about a spectrum.
- *SpectrumMinMax* – for keeping track of min/max amplitudes per wave number (and outputs these).
- *SpreadSheetRowGenerator* – generates spreadsheet rows from spectra, similar to the InstanceGenerator.

The following sinks are available:

- *SampleDataDisplay* – displays one or more sample data reports.
- *SpectrumDisplay* – displays one or more spectra.
- *SpectrumImageWriter* – writes a spectrum as an image.

Chapter 3

Visualization

The *Spectrum Explorer* (see Figure 3.1) allows you to load spectra of various types from disk or database for visual inspection, as well as processing via filters. PCA and PLS analysis are possible as well. The spectra can be exported any time as well. Actions in the GUI can be recorded as scripts and the replayed easily.

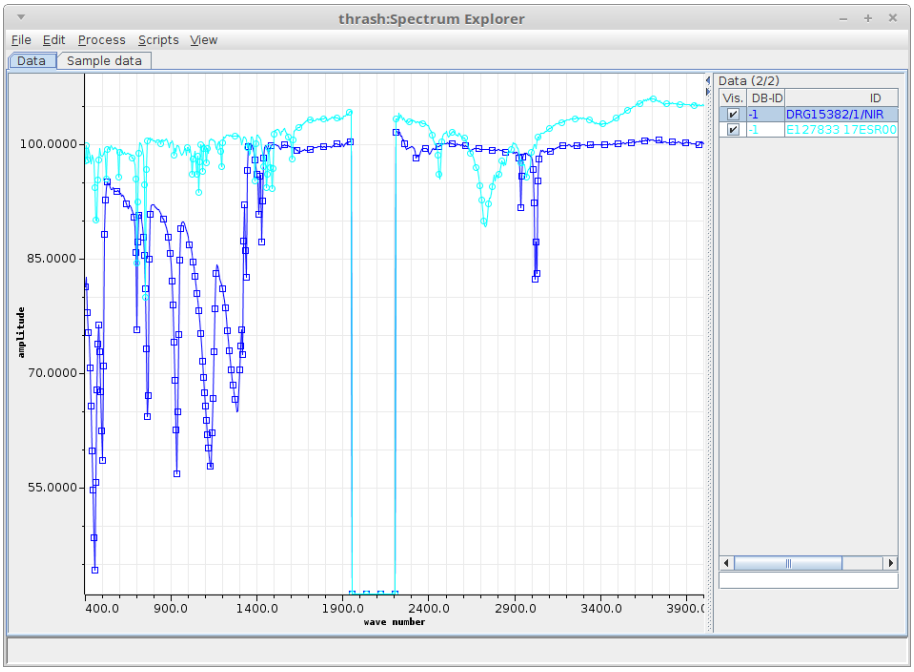


Figure 3.1: Spectrum Explorer

Chapter 4

Maintenance

The following maintenance tools are available:

- *Delete spectra* – allows removal of spectra from the database.
- *Delete sample data* – interface for deleting specified sample data values in selected spectra.
- *List sample data* – interface for listing all sample data values in selected spectra.
- *Update sample data* – interface for updating sample data values in selected spectra.

Chapter 5

Syntax highlighting

If you want to view .spec files, you can do that with basically any text editor, as they are just plain text files. However, there is basic syntax highlighting available for the following editors:

- vim
  `https://github.com/waikato-datamining/adams-spectral-base/tree/master/adams-spectral-2dim-core/src/site/resources/vim`
- Notepad++
  `https://github.com/waikato-datamining/adams-spectral-base/tree/master/adams-spectral-2dim-core/src/site/resources/notepadpp`

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
`https://adams.cms.waikato.ac.nz/`

[2] *NIR* – Near-infrared spectroscopy
`https://en.wikipedia.org/wiki/Near-infrared_spectroscopy`

[3] *MIR* – Near-infrared spectroscopy
`https://en.wikipedia.org/wiki/Mid-infrared`

[4] *XRF* – X-ray fluorescence
`https://en.wikipedia.org/wiki/X-ray_fluorescence`