

Unmixing with Custom Libraries

Peter Mason, June 2019

Introduction

A spectral unmixing algorithm determines mineralogy from reflectance spectra. Give the algorithm a rock's spectrum and it will tell you the minerals in the rock, quickly and all in a tidy package of results. It sounds too good to be true doesn't it? Before long, most spectral geologists get to see the technique fall on its face and realise that it has its limits. Some dismiss it brusquely while others look on it with slitty-eyed admiration, seeing its benefits even if it gets minerals wrong sometimes, and wondering if there's something they can do to improve it.

An obvious limit in TSG's unmixing support is the built-in spectral libraries. The current SWIR library has just 52 common minerals¹ and artefacts, and the TIR library has 102. An unmixing algorithm *cannot* report a mineral that isn't in the library, and may instead report the wrong mineral if it has a spectrum that's not too unlike the analyst's 'unknown' one. It doesn't take long for some analysts to ask about using their own libraries for unmixing. TSG has actually had full-blown TSA unmixing support for custom libraries for a couple of years now, but that hasn't helped much. It is a *mission* to put together a viable spectral library.

The latest release of TSG8 has functionality that allows you to **extend one of TSG's built in libraries** with some spectra of your own and use the combined library for unmixing. Your library could be quite elaborate, containing many new and / or replacement mineral classes – *or it could be very simple, containing just one or two new mineral spectra*. This document goes through the ground rules for putting together an extension (or full) spectral library, then explains how to use the library to generate unmixing results.

¹ The organisation of the SWIR library isn't entirely straightforward. 52 basic minerals (and artefacts) are reported but there are 60 classes at the unmixing level. (In some cases, e.g., chlorites and white micas, there is more than one unmixing class per mineral.) There are 493 individual spectra as in all cases there is more than one spectrum per unmixing class.

Ground rules for an external library

Here's a summary; details follow.

- The library has good-quality reflectance spectra of notionally-pure minerals.
- Some effort has been made to characterise the library samples.
- There is one spectrum per unmixing class for a TIR-style unmixing library, or a *cluster* of spectra per unmixing class for a SWIR-style library.
- Clones are forbidden. No two spectra in the library are the same.
- The spectrum names are suitably marked up so that TSG can relate them to its 'master' minerals table, and knows how they are organised.

Samples

Spectral unmixing goes beyond straightforward spectral matching and is normally done with spectra from notionally pure samples, not samples of mixtures. Ideally, the samples would be validated (e.g., using XRD and / or other techniques) so that their purity is actually known.

It can be very difficult indeed to source pure mineral samples and some leniency is normal, especially for SWIR spectra where some 'contaminant' mixed minerals (e.g., quartz) are basically invisible.

If you are able to get chemical data (elemental composition) for your samples then do so. Support for chemistry in TSG is growing and it may play a role in unmixing in the future.

Spectra

Try to get good-quality reflectance spectra. Use the best spectrometer available to you. Take care with the measurements, choosing the sample measurement area(s) carefully and using a long integration time. If at all possible, end up with absolute reflectance rather than reflectance relative to Spectralon (or similar).

Unique spectra

As noted no two spectra in the library may be the same. More generally, the spectra of two unmixing classes must be sufficiently different for the unmixing algorithm to be able tell them apart. If you can't see any differences between two class' spectral features then chances are the algorithm won't discriminate them either².

Library organisation and sample-name mark-ups

Ultimately, TSG will be extending³ one of its built-in libraries with your spectra so it needs some direction about how to handle them. Special tags are required in your library's spectrum names.

² Case in point: the lithium-bearing white mica lepidolite. There appears to be very little difference between the SWIR spectra of lepidolite and muscovite except in *some* cases lepidolite has a minor 1400 doublet. TSA has trouble discriminating between the two, doublet notwithstanding, and will easily report lepidolite in environments where there is no lithium.

³ Or if you happen to have a *complete* external library then TSG will be 'digesting' it like one of the built-in libraries. Either way, sample mark-up tags are required.

Spectrum ordering

Spectra do not have to be in any particular order in the library – they just need to be marked up correctly. TSG will sort them the way it likes.

TIR unmixing

I'll set the stage with an overview of how things are organised in unmixing. TIR unmixing is more straightforward than SWIR so let's start with that.

If you look at a TIR unmixing result (scalar set) in TSG then two views are available: the mineral level and the group level.

- The **mineral** level is the detailed view and it *seems* to be where unmixing actually happens. You see items like 'actinolite', 'hornblende' and so on.
- The **group** level is a coarser view. TSG takes the mineral-level result and combines minerals into their groups. E.g., actinolite and hornblende are combined in the single 'amphibole' group.

Now unmixing is actually done at a finer level than what you can see in the TSG mineral results. I'll call this **the unmixing-class level**. While there are 102 different minerals in the TIR library, there are 141 unmixing classes. For some minerals there's more than one unmixing class in the library. For example there are three 'hornblendes' in the library. Now I can imagine you saying: "Why use that affected term 'unmixing class' when you could just have said 'spectrum'?" I do apologise but you'll see why when we get to the SWIR. And yes, to be clear, for a TIR library unmixing-class level means library spectrum. To continue, the three 'hornblendes' (for example) in the library are different⁴. Each one gets a shot in the unmixing. If the algorithm happens to include two of them in its final result then the two will be combined afterwards to give a single 'hornblende' mineral fraction.

SWIR unmixing

SWIR unmixing in TSG is more complicated than TIR⁵. There is another, finer level of organisation which I'll call **the spectrum level**. While the TIR library has one spectrum per unmixing class, the SWIR library has at least two. The notion is for these spectra to span 'natural variation' in their unmixing class. The TSA algorithm uses this variation to construct a sophisticated 'CV' (canonical covariates) data space in which to do the unmixing. *The unmixing itself is done with unmixing-class means*⁶ but in this more-effective space.

Note Although TSG's built-in library has at least 2 spectra per unmixing class, you are allowed to bend this rule with your extension library. **You can introduce a few unmixing classes that have just one spectrum each.** Performance should not be badly affected if there are 2 or more spectra per unmixing class in the *majority* of the library.

⁴ We could debate about why but let's just accept that the library has three things called 'hornblende', each with a somewhat different spectrum.

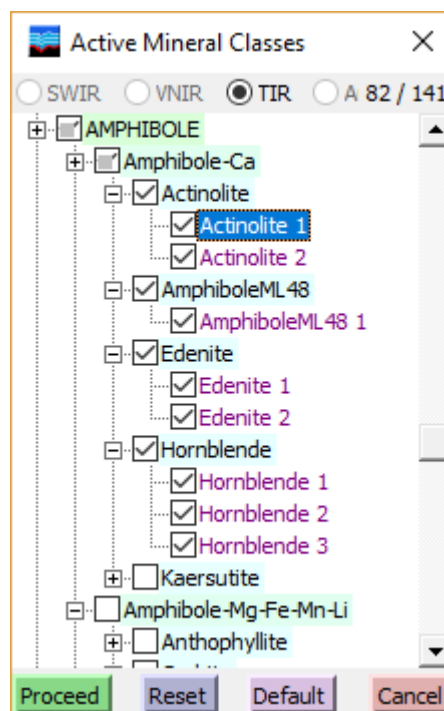
⁵ A big reason why things are organised more simply for TIR unmixing is that many more minerals have diagnostic spectra in the TIR than SWIR, and it is a challenge to find more than one relatively pure sample for each unmixing class.

⁶ Like with the TIR, only they are *mean* spectra here, not individual spectra.

Levels – redux

Below left is a table summarising how things are organised in the library and TSG. Below right is an example ‘Active Mineral classes’ selection list. You can see the AMPHIBOLE group, the Amphibole-Ca subgroup, some of the library’s amphibole minerals, and the unmixing classes (e.g., the three Hornblende ‘unmixing classes’ which are actually library spectra as this example shows a TIR library).

| Level | Spec range | Comes from | Seen in TSG |
|-----------------------|----------------------|---------------------|--|
| Spectrum | SWIR (and VNIR) only | Library | Never |
| Unmixing class | All | Library | Only in ‘active’ lists, in Settings and domain editing |
| Mineral | All | TSG, after unmixing | Scalars , Spectrum-screen overlay plot |
| Subgroup | All | TSG, after unmixing | Only in ‘active’ lists presently |
| Group | All | TSG, after unmixing | Scalars |



Library sample-name mark-ups

Note Mark-ups are separated by 2 spaces. We want something like this for a SWIR library sample:

Ankerite TableIndex:115 ClassIndex:41

And something like this for TIR:

Marialite TableIndex:36

...or maybe the luxury version with relative atom counts:

Marialite TableIndex:36

Chemistry:Na0.6437Al0.9527Si1.9181K0.0493Ca0.1969S0.0660Cl0.1785Fe0.0027O5.9921

TableIndex

To hook your library spectra up to TSG, each one needs a TableIndex tag in its name. e.g., **TableIndex:102** tells TSG that the sample is Epidote.

The indices for TSG’s known minerals are in this table:

https://research.csiro.au/static/tsg/tsg_minerals_list.csv

For each spectrum in your library, look up its mineral in the table and use the index that you find in a TableIndex tag, in the spectrum’s name.

If ‘your mineral’ does not appear in the table then contact me (peter.mason@csiro.au) to see about having it added.

ClassIndex

This tag is only used in SWIR libraries and indeed is required by them. It lets you control how the spectra are assigned to unmixing classes, and allows you to have more than one unmixing class per mineral if you want. Hopefully the SWIR chlorite example below will clarify this.

Invent ClassIndex values for your spectra, noting:

- The only *technical* requirement is to have different ClassIndex values for different unmixing classes of the same mineral. So in the chlorite example below, the ClassIndex values *could have* gone from 0 to 1 for Chlorite-Fe, 0 to 3 for Chlorite-FeMg, and 0 to 1 for Chlorite-Mg.
- A clearer approach is to have a different ClassIndex value for each unmixing class through the whole library. The system SWIR library is done this way and its last ClassIndex value is 59. You could start your new⁷ unmixing classes off at 60.

Blacklist

You wouldn't *normally* use this yourself, probably, but it's used in the system libraries so here's the story. This tag can be used to disable an unmixing class from automatically being active in unmixing. The analyst has to turn the class on manually, using the list in Settings[TSA] or the Domain editor. The default is **Blacklist:0**. Specify **Blacklist:1** to make the class inactive in unmixing unless manually enabled.

e.g., **Vonsenite TableIndex:132 Blacklist:1**

Unless blacklisted, Vonsenite seems too commonly found in TIR unmixing. Some mineral spectra are like that – they improve the curve fit in unmixing too 'easily' and should only be allowed to participate when there's a fair chance that they might be present.

Chemistry

This information doesn't have an active role in unmixing yet, but it probably will before too long. Supply it if you have it. What's asked for here is the sample's elemental breakdown **in relative atom counts**.

e.g., **Apatite TableIndex:130 Chemistry:Ca5P3O12Cl**

Comment

If you like you can include some general information in a Comment tag, e.g., a short description of the sample and / or its provenance. If present, comments are displayed in 'tooltips' for the active-class-selection lists. The Comment tag must be the last item in the spectrum name.

e.g., **Apatite TableIndex:130 Chemistry:Ca5P3O12Cl Comment: chlorapatite**

Example – Quartz in TIR library

You might expect that quartz would be the 'poster boy' for TIR reflectance but... well, maybe it is, actually. We can often see that classic M-shaped spectrum getting knocked around. A fairly common complication, I understand, is due to 'volume scattering' in small crystals. You might contact Andy Green if you'd like to know more.

⁷ If you want to boost one of the system library's (existing) unmixing classes with some extra spectra instead of adding new unmixing classes then it *is* possible but you might have some questions. Contact me (peter.mason@csiro.au) if you need help with this.

We currently have *six* quartz spectra in the TIR library⁸. As this is a TIR library, each one is an unmixing class. The TIR unmixing algorithm is allowed to find a solution containing more than one quartz component, and if it does so then TSG will combine the components' weights to report a single quartz mineral fraction.

```
Quartz TableIndex:8 Comment:Quartz crystal aggregate Sth. Dakota U.S.A. Pure (visually)
Quartz TableIndex:8 Comment:Quartz-Powdered 21:PQ3 Cook Sand
Quartz TableIndex:8 Blacklist:1 Comment:Quartz_A crystal Pure (visually)
Quartz TableIndex:8 Blacklist:1 Comment:Quartz_BC rock sample Impure (visually)
Quartz TableIndex:8 Blacklist:1 Comment:Quartz_F unknown
Quartz TableIndex:8 Comment:Quartz_LH unknown
```

Example – Chlorite in SWIR library

TSG's current SWIR library contains 31 individual chlorite spectra.

Chlorite is a common and important mineral and its composition is very meaningful to geologists. You will find that TSG reports **three chlorite 'minerals'**⁹ for the SWIR: Chlorite-Fe, Chlorite-FeMg¹⁰, and Chlorite-Mg.

In reality there are more than three possible Fe/Mg variations. A restriction that we apply in SWIR (but not TIR) unmixing is that the algorithm is not allowed to report two or more chlorites¹¹ in the same match. Also, the algorithm works by curve fitting. Imagine a real-world spectrum that is 'in between' our Chlorite-Fe and Chlorite-FeMg mean spectra. The algorithm would have to pick one to match it to, and it might not be such a great match. So then, to balance clutter (no more than three chlorite 'minerals' reported) and performance (better library matching to real-world spectra), we actually have **eight chlorite unmixing classes** in the library: two for Chlorite-Fe, four for Chlorite-FeMg, and two for Chlorite-Mg. This is the level where unmixing (and its curve fitting) happens, so having eight items (instead of three) covering the chlorite chemistry spread gives us a better chance of matching real-world chlorite well.

Here's how the sample names of the library's chlorites are marked up:

```
Chlorite-Fe      TableIndex:72 ClassIndex:18 Comment:Sample 1 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:18 Comment:Sample 2 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:18 Comment:Sample 3 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:18 Comment:Sample 4 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:19 Comment:Sample 1 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:19 Comment:Sample 2 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:19 Comment:Sample 3 of 4
Chlorite-Fe      TableIndex:72 ClassIndex:19 Comment:Sample 4 of 4
Chlorite-FeMg    TableIndex:73 ClassIndex:20 Comment:Sample 1 of 2
Chlorite-FeMg    TableIndex:73 ClassIndex:20 Comment:Sample 2 of 2
Chlorite-FeMg    TableIndex:73 ClassIndex:21 Comment:Sample 1 of 4
Chlorite-FeMg    TableIndex:73 ClassIndex:21 Comment:Sample 2 of 4
Chlorite-FeMg    TableIndex:73 ClassIndex:21 Comment:Sample 3 of 4
Chlorite-FeMg    TableIndex:73 ClassIndex:21 Comment:Sample 4 of 4
Chlorite-FeMg    TableIndex:73 ClassIndex:22 Comment:Sample 1 of 5
Chlorite-FeMg    TableIndex:73 ClassIndex:22 Comment:Sample 2 of 5
Chlorite-FeMg    TableIndex:73 ClassIndex:22 Comment:Sample 3 of 5
```

⁸ Although three of them are blacklisted as they aren't commonly encountered.

⁹ Our library has the common sort, not the more exotic ones like those with chromium.

¹⁰ aka 'Intermediate chlorite'.

¹¹ For example we don't allow SWIR TSA to report 60% Chlorite-Fe + 40% Chlorite-FeMg. This restriction is actually applied at the unmixing-class level. The algorithm is not allowed to report a result containing two chlorite unmixing classes, and there are eight of these classes.

| | | | |
|---------------|---------------|---------------|-----------------------|
| Chlorite-FeMg | TableIndex:73 | ClassIndex:22 | Comment:Sample 4 of 5 |
| Chlorite-FeMg | TableIndex:73 | ClassIndex:22 | Comment:Sample 5 of 5 |
| Chlorite-FeMg | TableIndex:73 | ClassIndex:23 | Comment:Sample 1 of 3 |
| Chlorite-FeMg | TableIndex:73 | ClassIndex:23 | Comment:Sample 2 of 3 |
| Chlorite-FeMg | TableIndex:73 | ClassIndex:23 | Comment:Sample 3 of 3 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:24 | Comment:Sample 1 of 4 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:24 | Comment:Sample 2 of 4 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:24 | Comment:Sample 3 of 4 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:24 | Comment:Sample 4 of 4 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:25 | Comment:Sample 1 of 5 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:25 | Comment:Sample 2 of 5 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:25 | Comment:Sample 3 of 5 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:25 | Comment:Sample 4 of 5 |
| Chlorite-Mg | TableIndex:74 | ClassIndex:25 | Comment:Sample 5 of 5 |

System library exclusions

Sometimes it's useful to be able to deactivate some of the unmixing classes in the system library that is being extended. For example, say you have a small extension library of epidote spectra and you would like to try them in place of the built-in epidote. Now you *can* turn off system-library unmixing classes interactively when generating an unmixing result, but it would be handy if the extension library could take care of this automatically. The good news is there is such a mechanism but the bad news is it's quite low-tech, quite user-hostile. Contact me (Peter Mason) if you would like to try it.

Summary

I'll finish this section with the main points.

- The TableIndex mark-up has an index into TSG's central minerals table.
 - This table can be downloaded here:
 - https://research.csiro.au/static/tsg/tsg_minerals_list.csv
 - You can't add minerals to the table. Contact me (Peter Mason) if you need to have a mineral added.
- For a **SWIR** library, each spectrum name includes TableIndex and Classindex mark-ups.
 - ClassIndex tags group spectra into unmixing classes.
 - There's normally more than one spectrum per unmixing class, but you can break this rule 'a bit'. You can have 'a few' unmixing classes that just contain one spectrum each.
 - There can be more than one unmixing class per mineral.
 - The unmixing algorithm is not allowed to report a result containing more than one unmixing-class item of the same mineral.
- For a **TIR** library, each spectrum name includes a TableIndex mark-up.
 - Each spectrum *is* an 'unmixing class'.
 - There can be more than one unmixing class per mineral.
 - The unmixing algorithm is allowed to report a result containing more than one unmixing-class item of the same mineral, but you wouldn't know as TSG will combine them to report one mineral-level fraction.

Using an external library for unmixing

I'll start with some discussion about putting a library together. TSG's new option for identifying an *extension* library is documented here¹².

Once your library (full or extension) is ready, the mechanics of using it to generate unmixing results are a little fiddly. I'll cover the important points.

Putting a library together

Spectrometer files

The usual situation is that you've done sample selection, validation and measurement, and now have a collection of files from the spectrometer (or its operating software rather). If they're common SWIR-spectrometer files then they're probably all alike and you can just import them in one go. Do a normal TSG import.

If they're from an FTIR spectrometer (or a configurable lab spectrometer) then you could be in for a bit of a ride. Configuration changes from one occasion to another can result in changes to the channels / wavelengths delivered by the spectrometer, and TSG won't put up with a mix when running a normal import. The way to go then is to use TSG's **dynamic import** functionality¹³. Start off by creating a TSG dataset from just one 'typical' file, choosing the desired resampling method and taking care that the import's wavelength range will accommodate all the other spectrum files. Then use the dynamic import system to add the other files to this dataset.

Wavelength resampling and subsetting

Given that a significant amount of work can go into preparing a spectral library, there is some appeal in having a 'master copy' TSG dataset of the library, perhaps at a higher resolution than what you think might be appropriate for unmixing.

Things to consider:

- If you are putting together a **full library** – not an extension library – then it's not the library's resolution that's in question here. Unmixing is done *at the production dataset's resolution*. When the time comes for unmixing, library spectra will be resampled on the fly to its bandpasses, as necessary.
 - The time taken for unmixing depends strongly on the resolution.
 - For example if you have a SWIR library sampled at 4nm and you use it to generate unmixing results for a dataset sampled at 1nm then the training¹⁴ and unmixing will take *quite* a lot longer than you might expect.
- If you are putting together an **extension library** then unmixing is done at the resolution of the built-in *system* library that you're extending. It's not necessary but I recommend that you prepare¹⁵ a library accordingly. The system resolutions are:
 - SWIR 1304 to 2496 nm with 4nm spacing

¹² If you have an extension library rather than a complete one then you have to mark it as special.

¹³ This functionality is described in a separate document.

¹⁴ TSA training is done just before the first unmixing calculation here, and it can take so much longer than 'usual' on a high-resolution library that you might think TSG has crashed.

¹⁵ You might prefer to do the resampling up front, with a method of your choice. You can use TSG's downsampler to do the resampling and / or spectral subsetting of your library.

- TIR 6000 to 14000 nm with 25nm spacing
- Later on when generating unmixing results with either a full or extension library, you will be able to reduce the active spectral range or exclude intervals¹⁶. It's just something to keep in mind.

Resampling method

Just a few recommendations here.

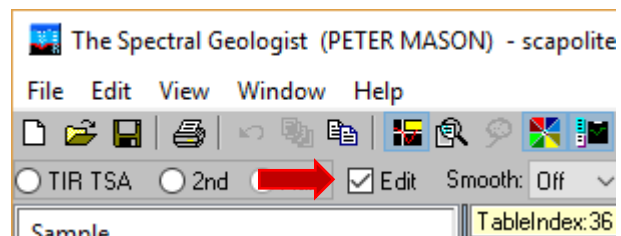
- When going from wavenumbers to wavelengths or vice-versa (e.g., FTIR spectra), the method of choice is Andy Green's 'Dyna L3' method. Gaussian bandpasses can be okay too, but can struggle badly if the sparse end¹⁷ of the input is too sparse.
 - This still applies if an FTIR spectrometer's spectra are *presented*¹⁸ with wavelengths rather than wavenumbers.
- When going from high resolution wavelengths to lower (e.g., binary ASD 1nm spacing to 4nm), the recommended method is **L3**.
- When dealing with identical wavelength spacing (e.g., both 4nm spacing) but different active ranges, the recommended method is **Linear** or L3.
- In difficult cases, for example spliced spectra with resolution steps, **Bandpass** is a good robust method.

Floater scratchpad

TSG's Floater Scratchpad can also be used for harvesting potential library spectra from datasets. In addition it can load ENVI spectral libraries¹⁹ directly and easily (drag & drop). Aside from the provenance of harvested spectra, the main thing to be mindful of here is spectral resolution. **The Floater scratchpad is kept at the resolution of the host dataset.**

Marking up sample names

Once you have your sample spectra in a TSG dataset, you need to mark up the sample names as discussed earlier. Open the dataset in TSG and switch to the **Spectrum screen**. Turn on the **Edit** checkbox. You can now work through the spectra, entering mark-up tags in the yellow edit field.



¹⁶ This is done when you 'attach' the library to a production dataset and is discussed below.

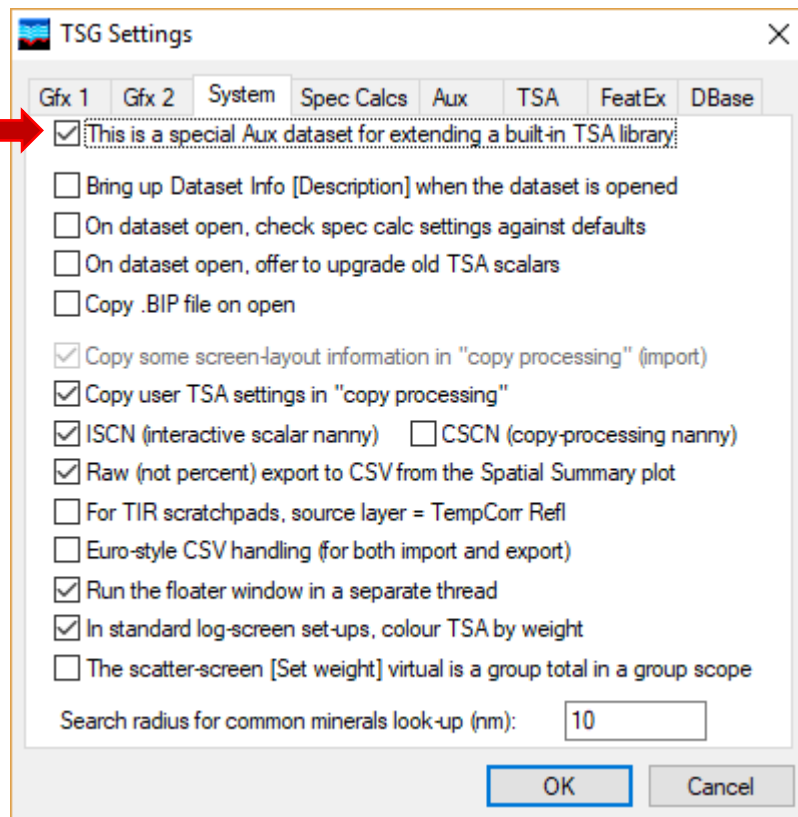
¹⁷ The side with small wavenumber values. FTIR spectra have constant or near-constant channel spacing in the wavenumber domain but this translates to variable spacing in the wavelength domain. For example in HyLogger3 TIR spectra, the channel spacing (in wavelengths) changes by an order of magnitude from one side of the spectrum to the other (from about 8nm to about 80nm).

¹⁸ This is done sometimes. The spectra are actually at their native resolution. If you look at the channel wavelengths then you will see a much smaller step on one side of the spectrum than on the other side. If you were able to see the channel wavenumbers then you would see a constant (or near-constant) step.

¹⁹ If you have a spectral library in TSG dataset format, it is easy to make it work like an ENVI spectral library too. Open the library dataset in TSG and then select the **File -> Export -> Write ENVI header -> Reflectance** menu. This will allow the library to be accessed as if it were an ENVI spectral library.

Marking an extension library

This step is for when you have an extension library, not a complete library.



Open your library dataset in TSG, then open the **File -> Settings** dialog and switch to the **System** page. At the top you should find a new checkbox called 'This is a special Aux dataset for extending a built-in TSA library'. **Turn it on.** Later on you will be attaching this library to production datasets in order to generate external unmixing results. When you do, TSG will know to do a special attachment. It will take the system TSA library that's most like your library in wavelengths, and extend it with your library's spectra. It is this extended library that will be used for the

external unmixing.

Where to keep aux libraries

Once you have attached an aux library to a production dataset, TSG will try to repeat the attachment automatically each time you open the production dataset. This goes on until you manually detach the aux dataset using the **File -> Detach aux** menu.

Now TSG does not remember full paths for aux datasets as this can lead to trouble when files are moved around. It remembers just the filename and looks for it in two places when trying an automatic aux attach:

1. Alongside the production dataset.
2. In the TSG Aux directory.

The 'TSG Aux directory'?

You can configure one or more directories as official aux-library directories for TSG. To do this, first create the directories as necessary, then create a Windows environment variable and enter your directory name(s) in it. If you have more than one directory then use the semi-colon as a delimiter.

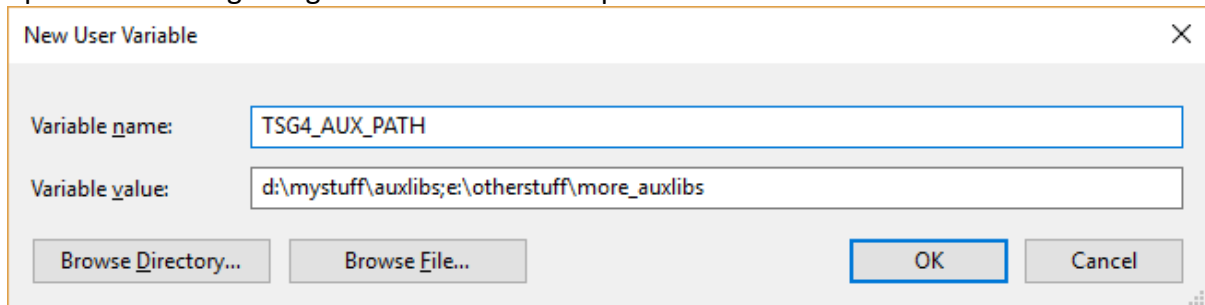
The environment variable must be named: **TSG4_AUX_PATH**²⁰

It can be a 'system' or 'user' environment variable.

For example in Windows 10, click your mouse over the Windows button at bottom left and type in 'Environment variable'. In my case, Windows comes up with two suggestions: 'Edit

²⁰ Yes, this mechanism has been around in TSG for some time.

the system environment variables' and 'Edit environment variables for your account'. Either one will do, although the second option is more acceptable by today's standards. In the dialog that you get, click **New...** to create a new variable called TSG4_AUX_PATH. Set up the next dialog along the lines of the example below.



You will probably need to restart TSG afterwards. You may even need to log off & back on.

Generating external unmixing results

In short:

1. Attach the aux library.
2. Visit the Domain editor for active-mineral subsetting (optional).
3. Generate an unmixing result set.
4. Fine-tune the mineral subsetting (optional).

Attach the aux library



With the production dataset open in TSG, use the **File -> Attach aux** menu to get a file-selection dialog. Select the Aux library's .tsg file.

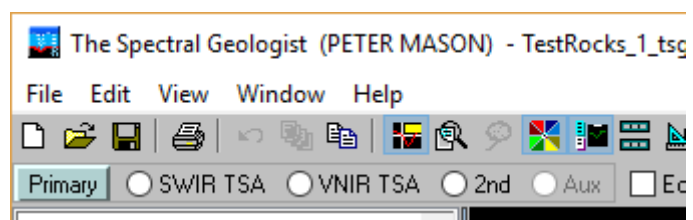
Alternatively, open a Floater window and just drag & drop the Aux library's .tsg file from Windows Explorer onto it.

Paired datasets

If you have a dataset pair (such as a HyLogger3 VNIR-SWIR + TIR pair) then it can actually have two concurrent aux attachments. TSG needs some help from you to handle things correctly.

If you are using the menu route then first switch to the **Spectrum screen** and click the **Primary / Assoc** button at top left until 'the right sort of spectrum'²¹ is shown.

If you are using the Floater route then first put the Floater in Spectrum mode () and toggle the dataset button () until 'the right sort of spectrum' is shown.



Warning

Do not change an attachment lightly. When you attach an Aux library to a production dataset, TSG hooks it up to the production dataset's domaining system. If a different Aux

²¹ e.g., a VNIR-SWIR spectrum for a SWIR library attachment.

library was attached before then its domaining info is first **purged** from the production dataset.

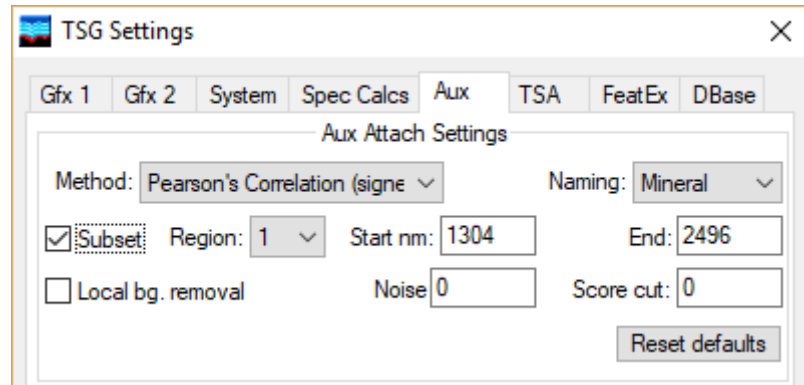
Clarification: VNIR-SWIR and TIR domaining info is managed separately. With a VNIR-SWIR + TIR production dataset **pair**, attaching a VNIR-SWIR Aux library will not affect any TIR aux domaining info.

Wavelength intervals

As soon as TSG has accepted an Aux library, it

automatically puts up the **File -> Settings [Aux]** dialog page so that you can check / change the wavelength intervals(s) active in unmixing.

Look out for the **Subset** checkbox.



- If it is on then take a close look at what the other settings are. Your TSG may still 'remember' traditional default SWIR aux-attach subsetting, where the 1400 and 1900 'water' regions are excluded. In general you would **turn Subset off**, only using it very deliberately for special unmixing runs.
- If it is off then *turn it on for a moment so that you can check the wavelength coverage*. This is more important for a full library than an extension library. Normally you'd turn it off again after checking.

Active mineral subsetting

Why

It is useful to be able to control which minerals in the library are active in unmixing. For example, maybe you have a library that includes a few 'exotic' minerals that you wouldn't expect to find routinely, so you wouldn't normally want to make them *available* to the unmixing algorithm. The idea is more general though. If you know that there is no geological sense in epidote (for example) occurring then turn it off to reduce chlorite misidentifications. Unmixing algorithms *do* make mistakes, and if you can restrict their opportunity to do so using your geological knowledge about the production dataset then you will improve the results.

Who should do this

This is for everyone.

As mentioned before, when you attach a properly marked-up Aux library to a production dataset, TSG hooks it up to the production dataset's domaining system²². This²³ is where active minerals and unmixing classes are selected for external unmixing.

²² TSG reserves an area here for a VNIR-SWIR attachment and another for a TIR attachment. I note this generally because it gives some background to the 'AUX S' and 'AUX T' items that all users will see in the Domain editor.

²³ not **File -> Settings [TSA]**

Analysts who work with core-logging datasets will most probably be familiar with domaining and RMS²⁴ lists, and the only new things here are the 'AuxS' and 'AuxT' RMS lists.

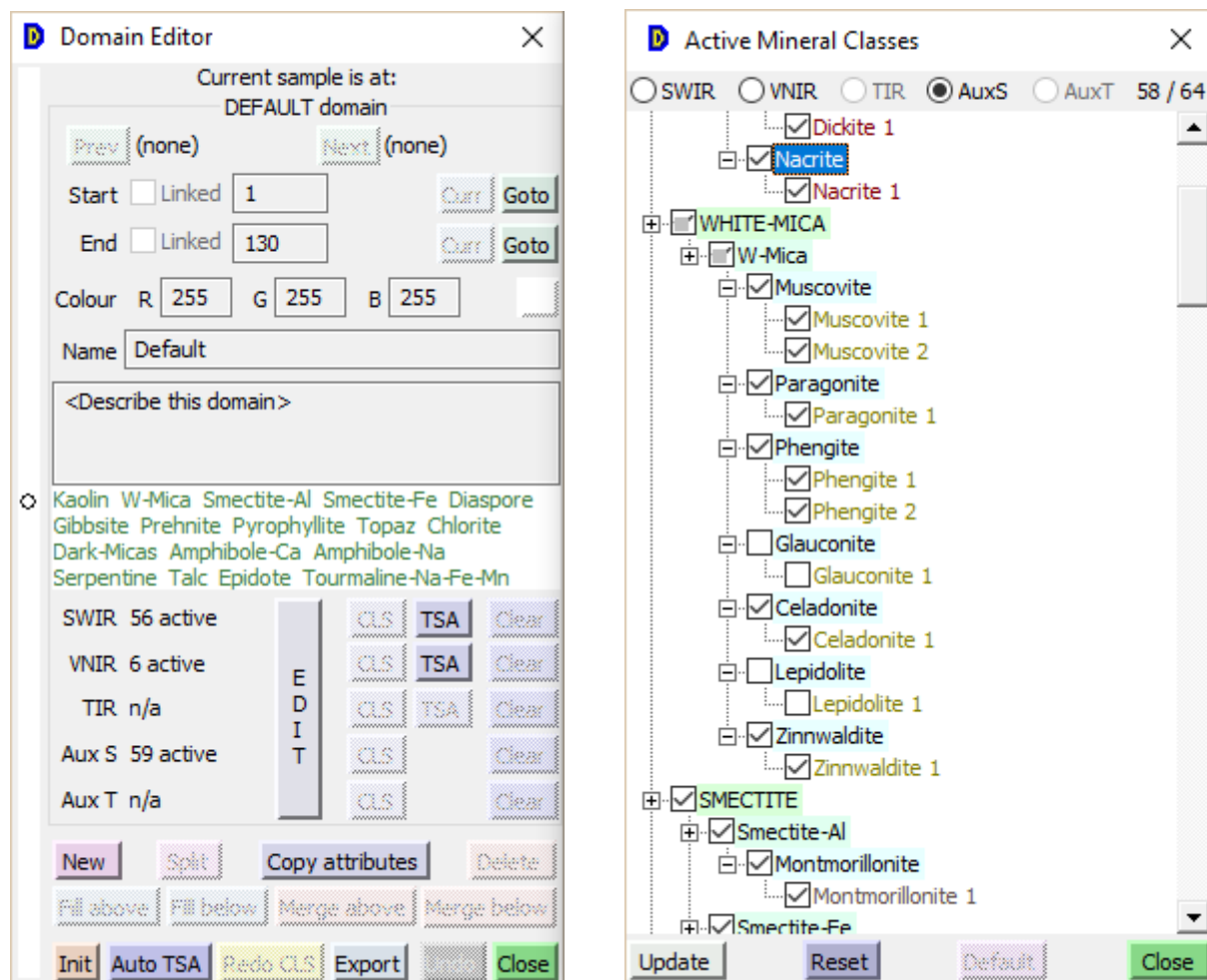
Analysts who work mostly with 'field' datasets probably have difficult questions like 'What?' and 'Why?' at this point. TSG has its peculiarities and *this* is how it does active-mineral subsetting for external libraries. Just go with the flow. You can ignore most of the domain editor's functionality. All that you need to do is edit the RMS list (active minerals list) for 'AuxS' or 'AuxT', for the 'Default' domain. Only one of AuxS and AuxT will be active, and every TSG dataset has a 'Default' domain automatically.

This section will cover new functionality and favour field-dataset analysts.

Domain editor

Bring up the Domain editor by clicking the **D** toolbar button.

You can visit the domain editor before generating unmixing results, and again later on to fine-tune the results.



The Domain editor's control panel is shown on the left. The list editor, shown on the right, is brought up by clicking the **E D I T** button.

The production dataset here is a VNIR-SWIR 'field' dataset. No domains have been defined for it – there is just the 'default' domain. That's probably how things will stay. We're here to edit the '**AuxS**' active minerals for the default domain.

²⁴ 'Restricted Mineral Sets', i.e., active minerals.

The list editor has five radio buttons along the top.

- The first three are for 'domained user TSA' results. We're not here for them so let's move on quickly before we get distracted.
- The next two are for external libraries. **AuxS** is for a VNIR-SWIR library and **AuxT** for TIR. This is a VNIR-SWIR production dataset and we have attached a compatible library so only **AuxS** is enabled. Select it to edit its list.

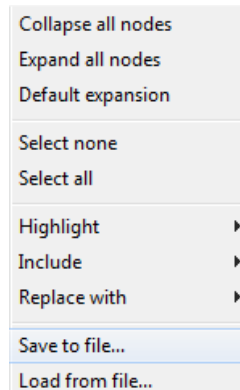
The list has four levels: group, sub-group, mineral, unmixing class. You can toggle items on and off at any level.

Click the **Update** button to commit any changes.

List dialog's right-click menu

The list editor has a **right-click menu**.

- The dialog allows one list to interact with another, e.g., the SWIR (conventional TSA) list with the AuxS list. The **Highlight**, **Include** and **Replace with** menus provide quick, convenient interaction but only work at the subgroup level.
- Using the **Save to file** and **Load from file** options, you can back up and restore your edits²⁵. A backup is recommended in case you change the aux attachment accidentally at some point and TSG purges the aux domain information. It also comes in handy if you like to experiment with different active mineral sets and / or different aux-library attachments.



Recalculate?

The **Redo CLS**²⁶ button goes live if you commit any edits to the list. If you have already created an unmixing result for your library (described below) then click this button to have it updated.

²⁵ It only works for the current domain, but if your dataset only has a 'Default' domain then this is nothing to worry about.

²⁶ This button's name is stuck in the past. The Domain editor was originally introduced to 'tame' TIR unmixing that was done using the CLS unmixing algorithm. Its scope has expanded since then.

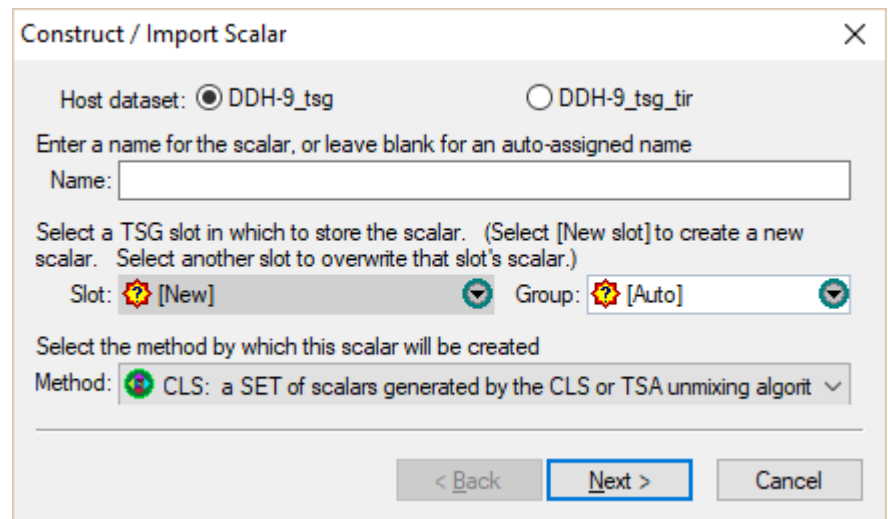
Generating an unmixing result set

Create a new scalar, e.g., using the **Edit -> New scalar** menu.

Select **CLS** in the **Method** list.

Note When working with a dataset pair you will find two '**Host dataset**' radio buttons along the top. *Select the right one.*

Click **Next**.



The 'Construct / Import Scalar' dialog box shows the following settings:

- Host dataset: DDH-9_tsg DDH-9_tsg_tir
- Name: [Empty text box]
- Slot: [New] [Existing]
- Group: [Auto] [Other]
- Method: CLS: a SET of scalars generated by the CLS or TSA unmixing algorithm

Buttons: < Back, Next >, Cancel

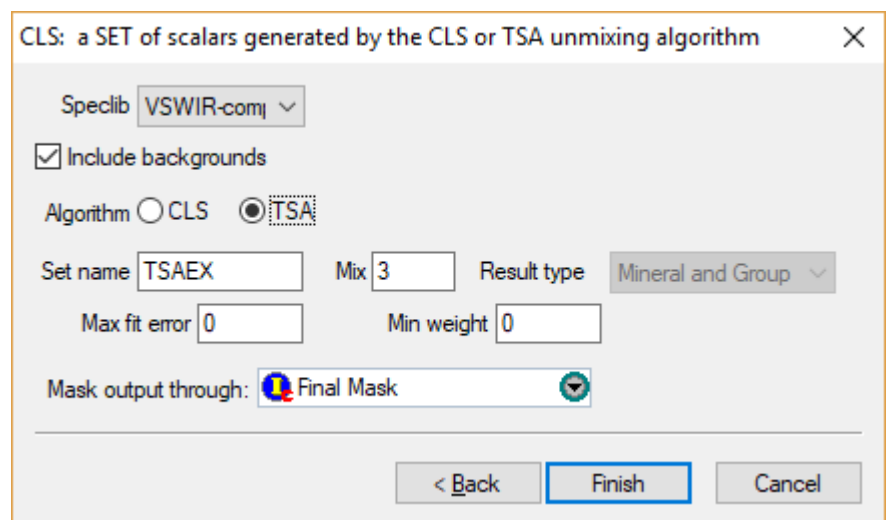
Look in the **Speclib** list for the aux-dataset's special entry. A VNIR-SWIR library will be called **VSWIR-compat. Aux** and a TIR library will be called **TIR-compat. Aux**. Select it.

Select the unmixing **algorithm** to use.

Enter something helpful in **Set name**.

Set the **mixture** level that you'd like the algorithm to generate.

Click **Finish**.



The 'CLS: a SET of scalars generated by the CLS or TSA unmixing algorithm' dialog box shows the following settings:

- Speclib: VSWIR-compat
- Include backgrounds
- Algorithm: CLS TSA
- Set name: TSAEX
- Mix: 3
- Result type: Mineral and Group
- Max fit error: 0
- Min weight: 0
- Mask output through: Final Mask

Buttons: < Back, Finish, Cancel

Imaginary FAQ

- How about I set **Speclib** to **File** and then go and select my aux-library dataset TSG file with the controls that appear. Can I do that?
Don't do that. That's an older, crude way of generating unmixing results. There is no control over the active unmixing classes, no TSA support, and TSG can't use the results as effectively.
- I've heard about a **jCLST** unmixing algorithm in TSG. Why don't I see it here?
Andy Green's jCLST algorithm is for TIR unmixing. It's only available when you have a VNIR-SWIR + TIR production dataset pair as it uses SWIR unmixing as a 'chaperone' for those minerals that respond better in the SWIR. (It also uses the VNIR spectra to moderate iron oxides.) jCLST *is* offered here when appropriate, and in such cases it uses the built-in SWIR library to the extent possible.
- What's this '**Include backgrounds**' option and can I turn it off?
When this option is on, standard background 'spectra' are added to your external library. These are necessary for the background modelling that the TSA unmixing algorithm *always* does, and the CLS algorithm *can* (and should) do. If you turn it off

then you won't be able to use the TSA algorithm. Turning it off is for advanced CLS unmixing.

- Tell me more about '**Set name**'.

This scalar construction method is unusual because it creates more than one scalar. When presenting page 1 of the scalar wizard I didn't mention entering something for the scalar **name**. TSG would just ignore it. The only control you have is the name of the **set**. TSG makes up its own scalar names, mostly, but your set name appears in these names.

Set names appear here & there in TSG. For example, the main selection list for a Summary-screen plot works with sets and the useful set-scope option in the Scatter screen also works with sets. It helps to have meaningful set names.

Another point – you aren't restricted to just one external unmixing result set.

Although you can only have one external library attached at a time, you could build up a number of result sets in a dataset. Only one would be 'live' but all the others would still be available for plotting. It is very helpful to have meaningful set names in this situation.

A final point – consider the bigger picture. If you take to external unmixing then it will help if you are consistent with your set names. If others receive your worked-up TSG datasets then this is even more important.

- Where do external-unmixing scalars go?

You'll find them in the **Assemblages and Auxmatch scalar folder**.

- Which **algorithm** and what **mix** level should I use?

For SWIR, the recommendation is TSA algorithm and a mix level of 2 or 3. SWIR TSA – done conventionally – has some library requirements but they should not be of concern if you have a reasonably small *extension* library.

For TIR, the recommendation is a mix level of *at least* 3, and the jCLST algorithm if possible. 6 is common. Good active-mineral subsetting is important too. If jCLST is not available because you don't have a dataset pair with a SWIR part then you will have to choose one of the others. Although the TSA algorithm is better than CLS, it is *significantly* slower at high mix levels²⁷ and with many active minerals, so CLS is the usual choice.

- What is **Max fit error** for?

It's one of the two quality-control filters offered for external unmixing. It's the maximum fitting error allowed. A sample gets NULL results if its best fit has a greater error.

For the **CLS** algorithm the fitting error is an RMS²⁸ error and if you pass in 0 then no error filtering is done.

For the **TSA** algorithm it is a scaled RSS²⁹ and if you pass in 0 then the corresponding TSA-library setting (typically 1000) is used.

- What is **Min weight** for?

It's the second quality-control filter. It is a minimum allowed *proportion* (fraction of 1) in an unmixing result. Tiny components in an unmixing result are viewed with suspicion.

²⁷ Without aggressive active-mineral subsetting, TSA with a relatively large library (like the TIR one) becomes *impossibly* slow at mixture levels beyond about 4.

²⁸ Root mean square error. This is usually well below 1, e.g., 0.1 is probably a bad fit.

²⁹ Residual sum of squares. Around 100 is usually a good fit and 1000 is the usual cutoff.

The **CLS** algorithm only computes one fit per sample. Any minor components with below-threshold proportions are stripped off, then the proportions of the survivors are rescaled to sum to 1. If **min weight** is 0 then this filtering is not done.

The **TSA** algorithm works in two stages. Thousands of candidate results are found in the first stage. The second stage judges the best³⁰ of these candidates, ultimately choosing just one or declaring a bad result. One of the items judged in a candidate is the smallest component it contains. If this component's proportion is below threshold then the candidate fails judgement and is discarded. If **min weight** is 0 then the corresponding TSA-library setting (typically **0.15**³¹) is used.

- TSA has like a million settings and I only see a couple here. What about the others? You exaggerate – it's more like four hundred and seventy thousand or so. For the time being they are copied from the production dataset's User TSA configuration.

Refining an unmixing result

I'll give some technical outlines here.

- How to modify or delete a result set that you have created.
- How to view residual spectra that relate to your result set.
- What can you do with your result set.

Modifying an external unmixing result

The main thing to modify is the selection of minerals active in unmixing. This is done using the Domain editor and has been discussed already. Reminder: after changing the **AuxS** or **AuxT**³² active-mineral selection, you can click the **Redo CLS** button in the Domain editor's control panel to have your results set recalculated.

More fundamentally, if you would like to change the mixture level, for example, or just force a recalculation, you can bring the result up for modification:

- Plot *any one* of the set's scalars in TSG, then select '**Modify scalar**' (somewhere) in the plot's right-click menu. For example, plot the '**min2**' (mineral2) scalar in a log-screen column, right-click the column and select '**Edit / Add / Del -> Modify scalar**'.
- You will get a normal scalar-construction wizard page. It'll look like the second picture on page 15 above.
- Make any changes you want and click **Finish**³³. The *whole set* is recalculated. If you reduced the **Mix** level then redundant scalars get deleted, and if you increased it then new scalars get created.

Complications

The scalar set can only be modified if the right aux library is attached.

TSG remembers which aux library goes with which scalar set and will find some way to make you uncomfortable if you try to modify a set when the wrong library (or no library) is attached. Things to consider.

³⁰ Normally a few dozen of the most promising candidates, determined by fit error at this point.

³¹ Keep this default in mind when using TSA at higher mixture levels. For example you might want a lower threshold when going for mixtures of 4.

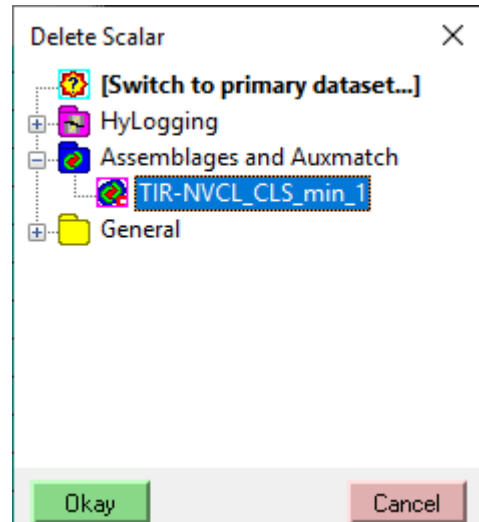
³² Whichever applies to your library

³³ **Finish** will cause the set to be recalculated even if you don't change any settings.

- When you open a production dataset that had an aux library attached before, TSG needs to be able to *find* the aux library. See the notes on page 10.
- Detaching and then re-attaching the *same* aux library is harmless, but you attach a *different* aux library then TSG will purge aux domain information. (TSG will warn you first.) You need to back-up / restore domain info *manually* to avoid loss of work. See page 13.

Deleting an external unmixing result set

If things just don't work out and you would like to expunge a result set from your production dataset then use the **Edit -> Delete scalar(s) -> Delete one scalar** menu to bring up the scalar deletion dialog. Yes, we're carrying on as if we were deleting just one scalar. It's 'one of those TSG things'. Under the **Assemblages and Auxmatch** folder the dialog will show (just) the mineral1 scalar of each custom unmixing result. Select the one you want to get rid of and click **Okay**. The entire unmixing result (all scalars and the set definition) will be deleted.



Residual spectra

Given an 'unknown' spectrum, the unmixing algorithm calculates likely component fractions from its library *and builds a model spectrum³⁴ from them*. Then:

$$\text{residual spectrum} = \text{unknown spectrum} - \text{model spectrum}$$

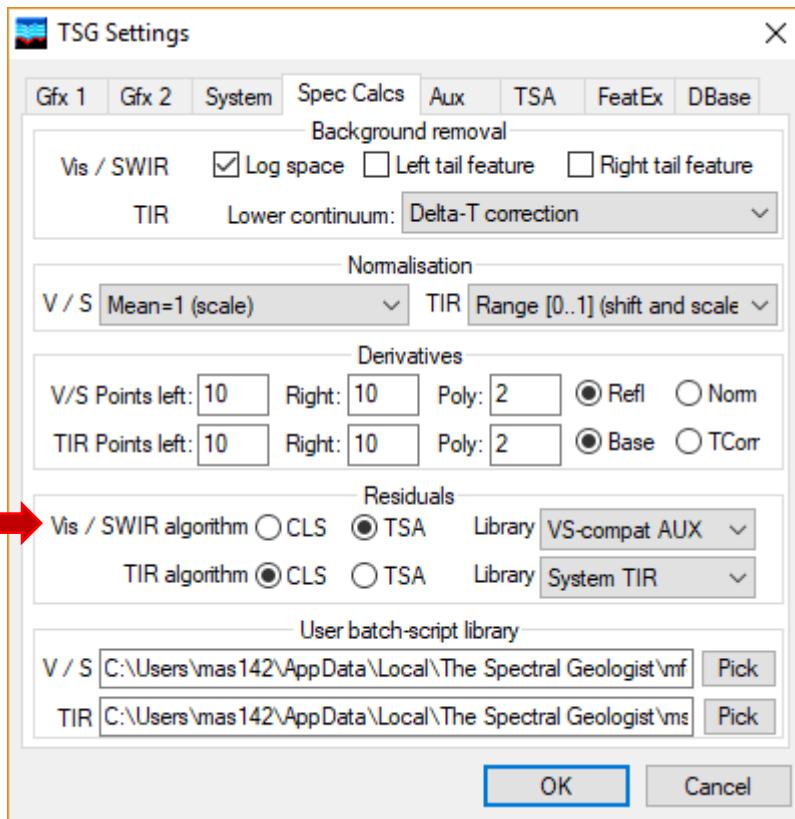
Residual spectra are useful for judging how well the unmixing algorithm has worked. You can see which features it didn't find, which features it sort-of found but not quite at the right place, and so on. With some knowledge and experience in working with mineral spectra, you will find residual spectra useful in fine tuning your library's active mineral set. The general idea – if you have the time – is to prune the active set down to just what is needed for the unmixing algorithm to model your production dataset's spectra well, leaving residual spectra that show little activity. The fewer unneeded library spectra you leave active in the library the better, as you are eliminating opportunities for the unmixing algorithm to make mistakes. Also, if there is an unfitted feature in some of the production dataset's spectra then your knowledge³⁵ should guide you in adding suitable library classes to the active set.

³⁴ If you have played around with the Floater's TSA mode then you will have seen such model spectra.

³⁵ General knowledge of mineral spectra along with geological knowledge of what minerals one would expect to find in the production dataset.

Configuration

Bring up the Settings dialog with the **File -> Settings** menu and switch to the **Spec Calcs** page.



About 2/3 the way down you'll find the controls for spectral residuals. (This example deals with a dataset pair so there are two lines of controls here for residuals.) Select the **algorithm**³⁶ you used for your external unmixing, then select your aux **library**³⁷ from the list. Note that if you use the TSA algorithm then you can only view residuals after creating a result set. On the other hand, you can view CLS residuals before calculating anything.

Viewing residual spectra

You can view residual spectra in the Log, Spectrum, Stack and Hole screens. Select the residual layer to plot. It is named after the algorithm selected in Settings. It will be called either **TSA Residual** or **CLS residual**.

The multiple-sample views provided by the Log and Stack screens are effective for revealing areas of *consistent* residuals. If you see a feature that has been missed (has a residual) over several samples, not just one, then you can be fairly sure that something's missing from your active mineral set (or library).



Note that the residual spectra will be zero over some of the wavelength range if the library does not span the production dataset's full wavelength range.

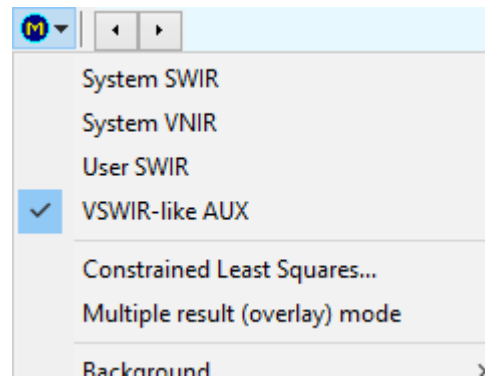
At this time the Floater Spectrum mode does not offer to plot the residual layer. However, the Floater's TSA mode makes up for this.

³⁶ Again, just go with the flow. 'This part of TSG' just needs to be told how you configured 'that part of TSG'.

³⁷ If your library is marked up properly then TSG will have integrated in its system during the attach process. Your library will then appear on the list, named either **VS-compatible AUX** or **TIR-compatible AUX**.

Floater TSA mode

If you have attached a properly marked-up library then you can use the Floater to plot TSA results for it. This will work even if you haven't created a result set yet. After attaching the library, start a Floater and put it in TSA  mode³⁸. Bring up the Floater's menu with a right click or via the  toolbar button. You should find a new TSA 'database' selection named **VSWIR-like AUX** (or **TIR-like AUX**). Select it and the floater will show you 'live' TSA results for your library. You can also use the Floater TSA's **Multiple Result mode** and **Constrained Least Squares** tool with a custom library.




Other support in TSG

To finish off, here's a brief mention of some other functionality in TSG that relates to unmixing result sets.

- The **Summary screen** works with scalar sets. Your result set is one of these. You can plot Overview, Spatial and Difference histograms for it. These plots can be at the group, mineral or element level.
- The **Log screen** has a **quick pre-set** for showing external unmixing results (called 'Custom set' here). You've probably seen the ones that set up log columns for the usual TSA results. Now there's one for your external unmixing results too. If you have more than one external-result set then it'll do them all.
- The **Scatter screen** has a **set scope** for each sub-screen. You can use your external result set here.
- There are two items of interest under the **Edit -> Scalar sets** menu.
 - **Copy a scalar set** lets you spin off copies of the set's main scalars³⁹.
 - **Play out a scalar set into scalars** works on it⁴⁰.
- The set can be presented as '**consolidated**' **Minerals** and **Groups** items in a few places in TSG (such as the Class Extraction scalar and the Downsampler). This is about treating the custom set as a single entity.

-oOo-

³⁸ If you have a dataset pair then you must first associate the Floater with the 'right' dataset, using the  toolbar button.

³⁹ You get a new set of import-type scalars. You can make some manual adjustments afterwards: The class scalars can be class-edited manually, and the Floater TSA's manual **Replace result in user set** option (documented elsewhere) works on the set.

⁴⁰ This tool expands a scalar set into one (new) scalar per group, subgroup, mineral or element.