Sparrow

Single PARticle Retrieval Of Waves

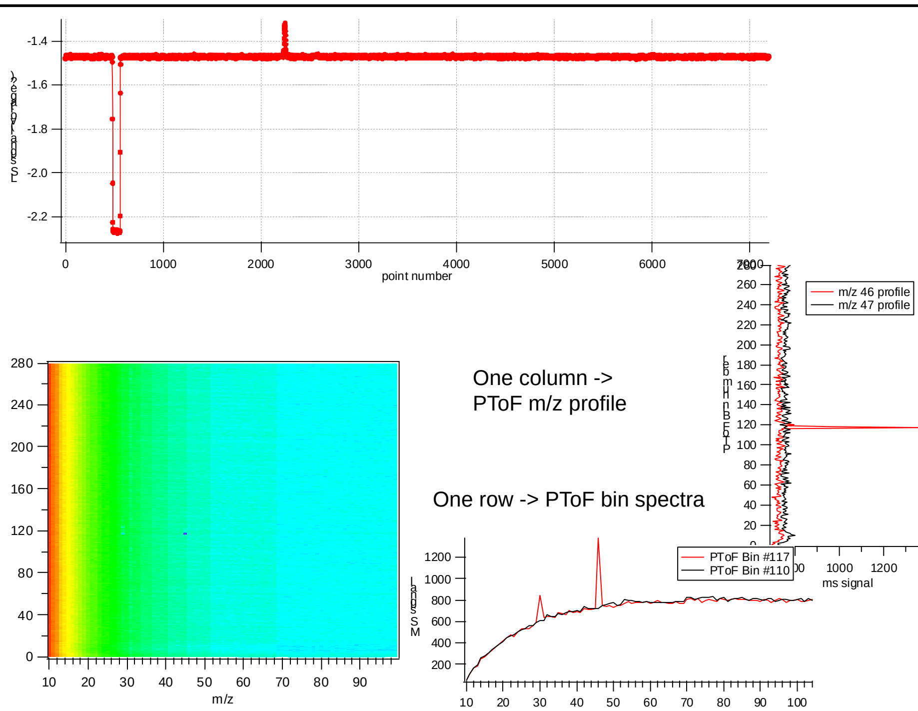
Light Scattering (LS) Analysis Tool

The first flight

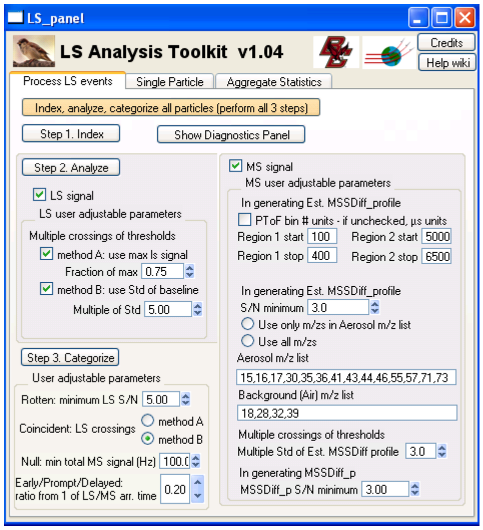
ver. 1.04

AMS Users Mtg, 2010

Eben, Tim, John J, Donna



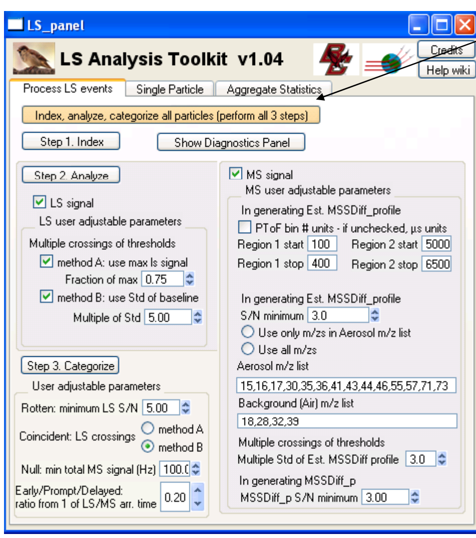
LS Analysis Tool Introduction



The main analysis is done in the Process LS Events tab and consists of 3 steps:

(1) Index
(2) Process
(3) Categorize

LS Analysis Tool Introduction

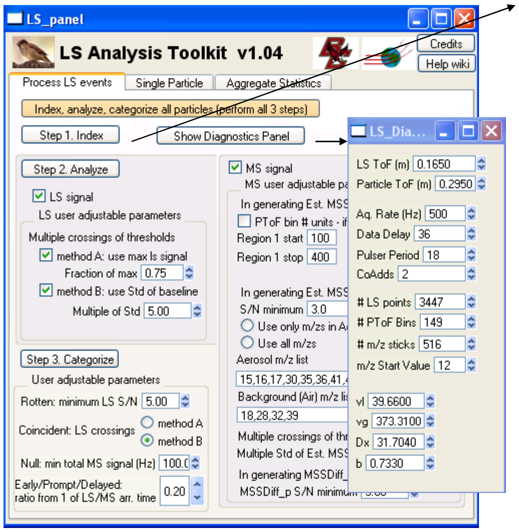


Typical users will always simply press the gold “Index, analyze..” button and use the defaults.

Once all the data has processed, one can go back and adjust any relevant parameters and perform individual steps on individual particles or todo waves.

Font legend for next several slides:
BlueAndBold -> a 1or 2D wave residing in memory, typically the LS folder, typically containing analysis results.
GreenAndBold -> a user-adjustable parameter on the panel that affects results

Step 1. Index



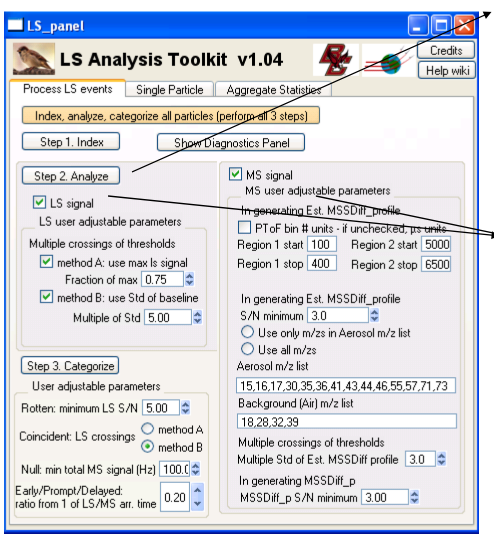
At the step the user is prompted for the folder containing the DAQ LS data files. These files are named yymmdd_xxxxxxxxx_lsSt.hdf.

Diagnostic values are loaded into waves and global variables are set based on the DAQ settings. Users can view and modify the global variables in the Diagnostic Panel.

So far the code only allows for analysis of LS data taken with the same DAQ settings (i.e min and max range of m/z sticks, PToF calibration values, etc.).

Along with global variables, the index step also creates some waves:
msChop_PTOF = 1e-6*(DataDelay + PulserPeriod*CoAdds*p), for plotting spectra data vs time
msChop_mz_amus = mzStartVal+p, for plotting spectra data vs m/z.

Step 2. Analyze



In this step we examine the light scattering (LS) signal and/or the mass spectra (MS) signal. We gather information about maximum values, etc. on each LS event (LS event = LS trigger = LS particle = particle run number). The user will be prompted for an intermediate data file path.

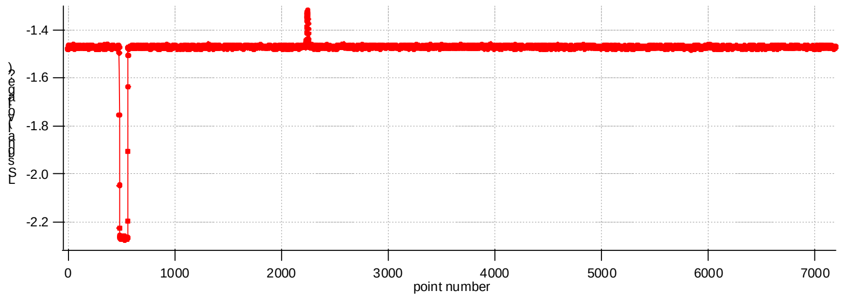
Users can optionally Analyze the LS and the MS data separately, but some results of the LS analysis is needed for the MS analysis.

When we are finished analyzing individual particles, we have a 2D matrix that is roughly analogous to the MSSDiff_p matrix that squirrel uses.

Font legend for next several slides:
BlueAndBold -> a 1or 2D wave residing in memory, typically the LS folder, typically containing analysis results.
GreenAndBold -> a user-adjustable parameter on the panel that affects results

Step 2. LS analysis

For each LS event we have a 1D light scattering wave called **LSChop**

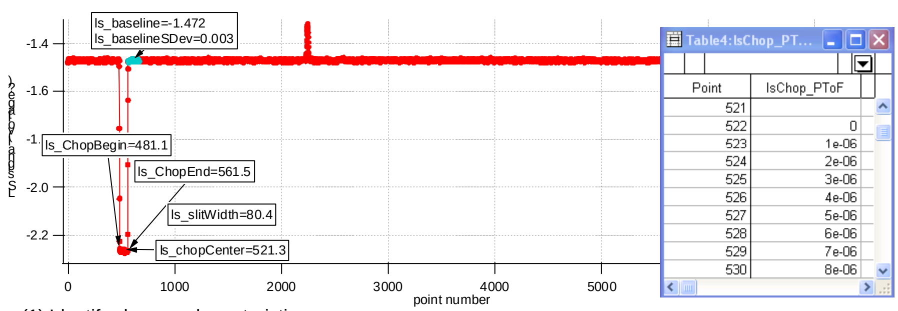


The main analysis tasks for each LS signal are:

(1) Identify chopper characteristics (start, center, end)
(2) Calculate a baseline and the standard deviation of a non-signal region.
(3) Translate the x-dimension from point number to time.
(4) Find the maximum LS signal and associated statistics.
(5) Estimate the number of particles seen in this single LS event (nominally1).

Step 2. LS analysis

For each LS event we have a 1D light scattering wave called **LSChop**



(1) Identify chopper characteristics

* Find point number of minimum LS value. Find point numbers before & after minimum value that exceeds minimum value by 0.5 (volts). Save to **ls_ChopBegin, ls_ChopEnd**.

* Find and save **ls_slitWidth** (ls_ChopEnd-ls_ChopBegin), **ls_ChopCenter** (ls_ChopEnd-ls_ChopBegin)/2,

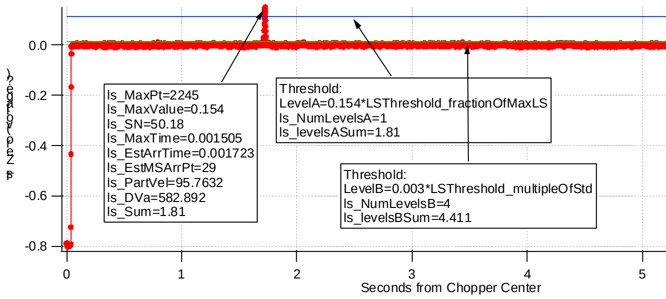
(2) Calculate the baseline and standard deviation

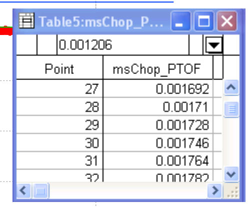
* Find the average, standard deviation of LS signal for the region Ceil(ls_ChopEnd)+5 to Ceil(ls_ChopEnd)+104. Save to **ls_baseline and ls_baselineSDev**.
* Subtract ls_baseline from LSChop, save to **LS_Zero.**

(3) Transform x dimension to time

lsChop_PTOF=(p-ls_ChopCenter)*LS_dt (LS_dt = 1/(kHzAcqRate*1000))

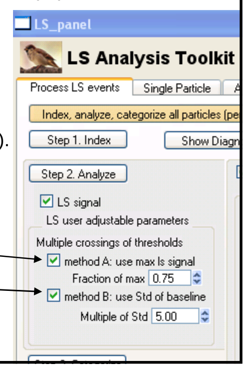
Step 2. LS analysis





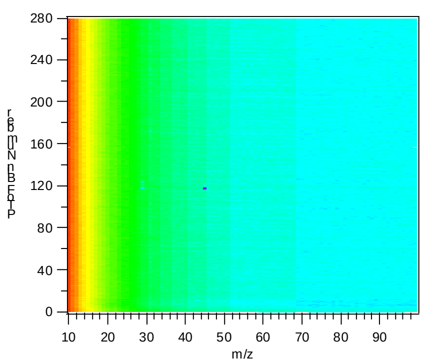
(4) Find the maximum LS signal and associated statistics.
Find **ls_MaxValue, ls_MaxPt, ls_MaxTime, ls_SN** (maximum signal to noise = ls_maxValue/ ls_baselineSDev). Using the ls_MaxTime, calculate **ls_PartVel**, (LToF_distance/ls_MaxTime) **ls_DVa** (using Vl, Vg, b, Dx), **ls_EstArrTime** (PToF_distance/ls_PartVel), **ls_EstMSArrPt** (point number in MS space of maximum LS signal), **ls_SumLS** (sum of LS_Zero over the range ls_MaxPt +-10).

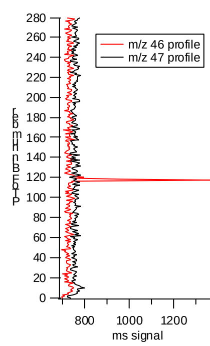
(5) Estimate the number of particles seen in this single LS event by two methods:
(a) **ls_NumLevelsA** = level crossings of ls_MaxValue***LSThreshold_fractionOfMaxLS**
(b) **ls_NumLevelsB** = level crossings of ls_baselineSDev***LSThreshold_multipleOfStd**
Find the min and max point positions of the level crossings, sums of LS_Zero:
ls_LevelsASum = sum(LS_Zero , V_min-10, V_max+10)
ls_LevelsBSum = sum(LS_Zero , V_min-10, V_max+10)



Step 2, MS analysis

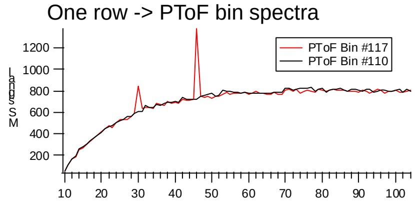
For each LS event we have a 2D mass spectra stick matrix called **sp_Stick**.





One column -> PToF m/z profile

One row -> PToF bin spectra

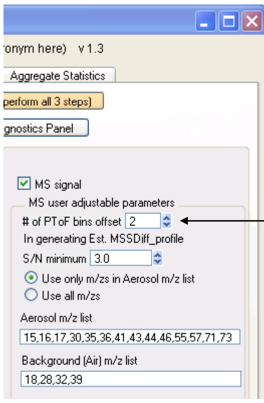


The main analysis tasks for the 2D MS sp_stick matrix are:

(1) Apply a DC-marker type of correction to get the signal with the zero removed
(2) Identify the PToF time in which the MS sees a particle
(3) Generate a single mass spectra

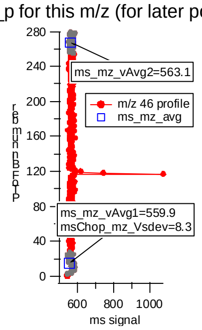
Step 2. MS analysis

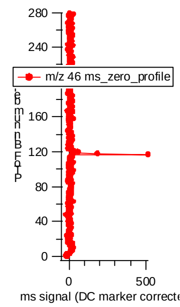
(1) Apply a DC-marker type of correction to get the MS signal with the background removed.



* Apply duty cycle correction to sp_Stick (*sqrt(28/mz)). It is not necessary to perform this correction at the very beginning; we simply do it first to get it out of the way.

* For each m/z extract the PToF m/z profile.
- Find the average and standard deviation of the first 25 bins beginning with **MSPToFBinOffset**; put result in **ms_mz_Vavg1**, **ms_mz_Vsdev**.
- Find the average of the last 25 bins, put result in **ms_mz_Vavg2.**
- Subtract the average of ms_mz_Vavg1, ms_mz_Vavg1 from the mz Profile; call this entity ms_zero_profile. Put ms_mz_zero_profile in ptof_stick_p for this m/z (for later possible squirrel use).





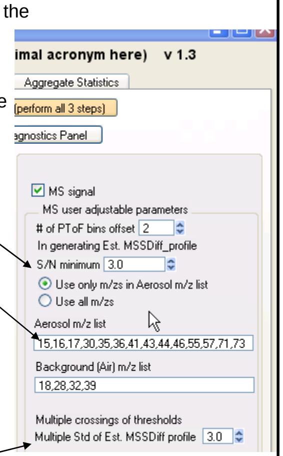
Step 2. MS analysis

(2) Identify the PToF time in which the MS sees a particle.

To accomplish this task, we generate a 1-D wave called **MSSDiff_profile**. MSSDiff_profile is the 2D matrix sp_Stick that has been collapsed in m/z dimension. See next slide for graphics.

For each ms_mz_zero_profile find the maximum value after MSPToFBinOffset, **ms_mz_Vmax**, the bin number of the maximum value, **ms_mz_VmaxLoc,** and the signal to noise **ms_mz_SN** (ms_mz_Vmax/ms_mzVsdev).

If ms_mz_SN > **MSThreshold_multipleOfStd**
(a) if considering only m/zs in **mzAerosolList** add to MSSDiff_profile
(b) else don't screen for mz aerosols, just add to MSSDiff_profile.



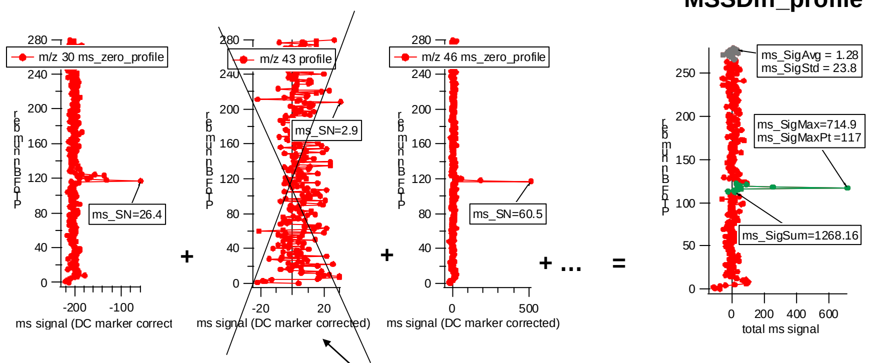
* Using MSSDiff_profile (a sum of baseline subtracted profiles of various mzs)

Find the average and standard deviation for the last 25 PToF bins; put results in **ms_SigAvg, ms_SigStd.**

* Find the max value, **ms_SigMax**, the maximum PToF bin number **ms_MaxPt ,** and the signal to noise, **ms_SigMaxOverStd** (ms_sigMax/ms_sigStd).
* Find the time of the maximum MS signal via ms_SigMaxPt and msChop_PTOF, **ms_SigMaxTime**.
* Using the ms_SigMaxTime and the PToF_distance, find the particle velocity **ms_PartVel.**
* Estimate overall signal at max value, **ms_SigSum,** (sum of MSSDiff_profile over the range ms_MaxPt +-5).
* Estimate the number of particles seen in this single LS event by level crossings of MSSDiff_profile by ms_SigStd***MSThreshold_multipleOfStd**

Step 2. MS analysis

MSSDiff_profile



The ms_mz_zero_profile for m/z 43 is not added into MSSDiff_profile because the signal/noise (2.9) is less than our threshold (default is 3 and it is not at the correct location in PToF space).

Table0:msChop_PT…

3.6e-05

Point	msChop_PTOF
116	0.004212
117	0.004248
118	0.004284
119	0.00432
120	0.004356
121	0.004392
122	0.004428
123	0.004464

ms_MaxTime = 0.004248

Note that in the calculation of MSSDiff_p (task 3) the position of the particle, ms_MaxPt, is used to screen an mz_zero_profile to determine whether any signal from this m/z is significant (i.e. shows up in this particle).

Step 2. MS analysis

(3) Generate a single mass spectra

We generate a 1-D mass spectra that we will put into the data set MSSDiff_p. This is the 2D matrix sp_Stick that has been collapsed in PToF bin dimension.

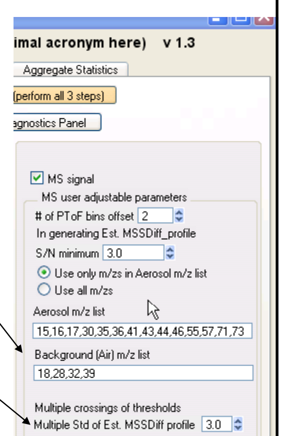
For each m/z examine the ms_mz_zero_profile.

If this m/z is in the Background m/z list, ignore.

Find all level crossings of ms_mz_zero_profile by ms_mz_Vsdev *MSthreshold_MSSDiff_mult_std

Compare the position of the max signal from the MSSDiff_profile, ms_MaxPt, to the location of level crossings.

If any of the positions of level crossings are within 2 PToF bins from the ms_MaxPt, find the sum of ms_mz_zero_profile within +- 2 bins of ms_MaxPt. Put this sum in the m/z position for this 1-D spectra.

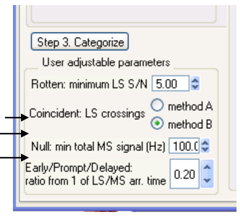


Step 3. Categorize

Every LS event is assigned one of 6 types:
- **rotten** (don't see a significant LS signal)
- **coincident** (more than one particle was detected in one LS trigger using LS signal)
- **null** (don't see a significant MS signal)
- **early** (time of max MS signal << time of max LS signal)
- **delayed** (time of max MS signal >> time of max LS signal)
- **prompt** (MS and LS signals line up in time reasonably well)

Three user-adjustable parameters in the LS data folder and their defaults are:

categorize_multipleStd 3 (the S/N of the max LS value must exceed this value)
categorize_minSig 100 (The sum of all aerosol species in units of Hz)
categorize_minRatioMS 0.2 (ratio of MS to LS away from 1, unitless)

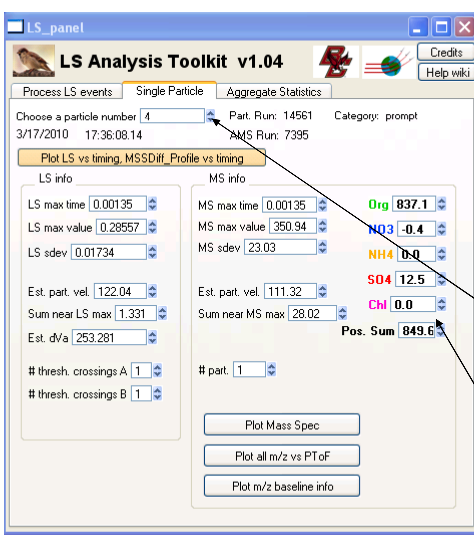


We use the ls_MaxTime to find average and std of LS timing for all particles.
Next, find these values: ls_mid = avg; ls_up = avg+3* std; ls_down = avg-3* std

For each particle let x = timing in PToF space of the max LS signal for this LS event
- * If x > ls_up or x < ls_down or LS S/N < categorize_multipleStd then event is **rotten**
- * If number of levels found from LS_step 13a > 1 then event is **coincident**
- * If summed signal in Hz from all aerosol species < categorize_minSig then event is **null**
- * If ratio of timing of particle arrival from MS to LS > 1 + categorize_minRatioMS LS the event is **delayed**
- * If ratio of timing of particle arrival from MS to LS < 1 - categorize_minRatioMS LS the event is **early**
- * If none of the above, the particle is **prompt**

Lastly, we calculate some diagnostic waves comparing the MS arrival time.
EstArrTimeMSLS_ratio = ms_MaxTime/ls_EstArrTime //Ratio of MS to LS arrival time
EstArrTimeMSLS_diff = ms_MaxTime - ls_EstArrTime //difference

LS Analysis Results



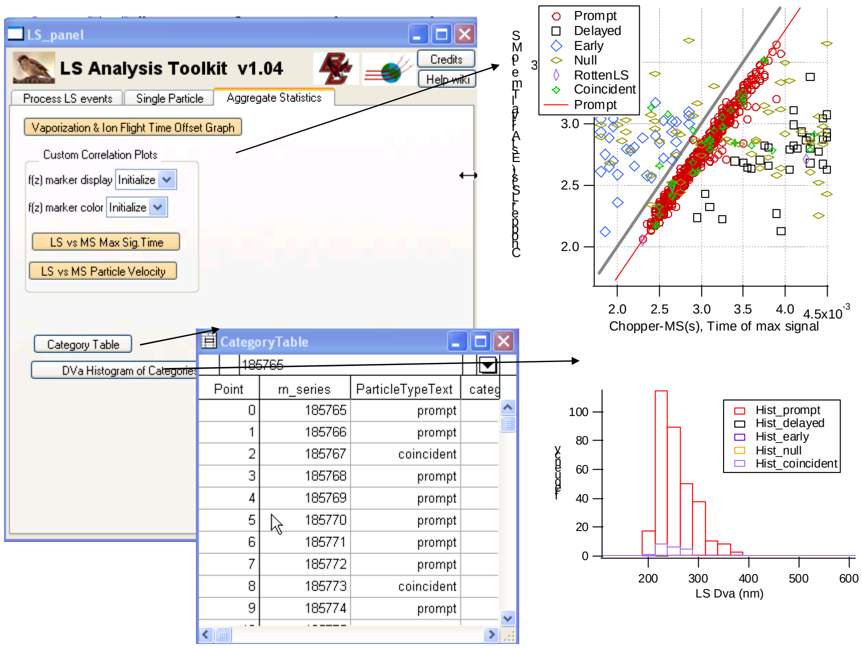
The results of the analysis can be viewed on a single particle or in aggregate.

The utility of viewing the analysis results of a single particle is to verify calculations, examine user-adjustable values, etc.

Particles are identified by their number (simply the row entry) or by the 'particle run' value.

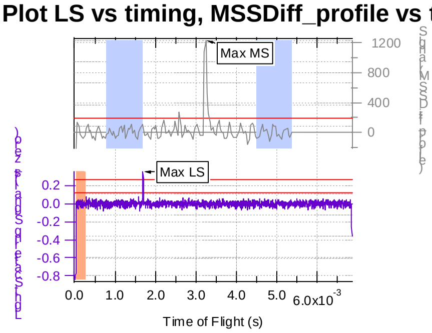
Many results are simply viewed as variable displays on the panel.

LS Analysis Results – Aggregate Statistics.

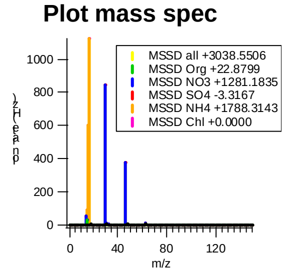


LS Analysis Results – Single Particle graphs.

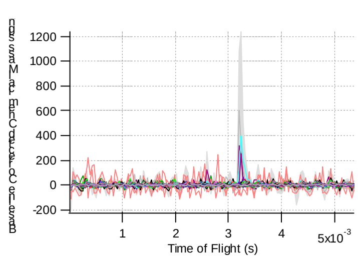
Plot LS vs timing, MSSDiff_profile vs timing



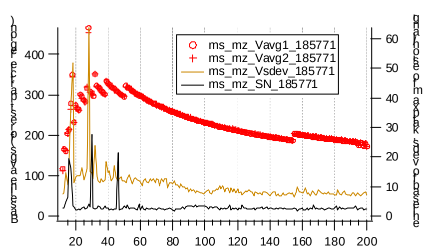
Plot mass spec



Plot all m/z vs PToF



Plot m/z baseline info



Some caveats about LS Tool version 1.4

• Some of the important squirrel panel tools work.

• Currently there is not a capability to perform and LS analysis when important DAQ settings such as the number of m/z per spectra or the number of PToF bins have changed between DAQ hdf files.

• One may use many tools in the MS tab, particularly the time series and average mass spectra on any todo wave.

• One may generate PToF size distributions, PToF image plots, and PToF average mass spectra plots on the all todo wave. Performing some of these operations on some non-all todo waves may not work because once again squirrel expects some values to be present for all particles, and the DAQ only saves some values once per file (parVal vl, vg, dx, b, etc). The x-axis for image plots currently corresponds to particle number (instead of time). This will eventually change.

• The Aggregate Statistics section is quite minimal, but will grow as the experience of the community evolves. User suggestions and contributions are VERY welcome! Most of the current tools have to do with generating basic statistics about the data and insuring that the instrument was working properly.