

Expand the information gained from your NMR data.

Once NMR data is acquired the real work begins.

Processing that data to coax out the valuable information users need is where the value of NMR technology is realized. Green Imaging Technologies' **NMRProLab** software provides access to a list of patented NMR data processing tools that can maximize the value of the data collected from time domain NMR instrumentation.

Presented in an easy to learn, customizable user interface, **NMRProLab** allows researchers to look deeper into their data to answer more questions and solve more problems.

**NMRProLab** provides access to the following NMR processing options:

- **1D continuous distributions:**
  - Free Induction Decay (FID)
  - CPMG T2
  - Inversion Recovery or Saturation Recovery CPMG
  - Diffusion distributions.
- **2D continuous distributions:**
  - T1-T2 from CPMG Inversion or Saturation Recovery
  - T2-Diffusion
  - T2-Store-T2
- **3D continuous distributions:**
  - T1-T2-Diffusion from CPMG Inversion or Saturation Recovery
  - Diffusion
- **1D, 2D and 3D Fourier transforms for imaging data**
  - Exponential smoothing.
  - Pixel interpolation
  - 3D Free form viewer with surface rendering
- **100s of other built-in functions that can be utilized in user defined processing routines.**

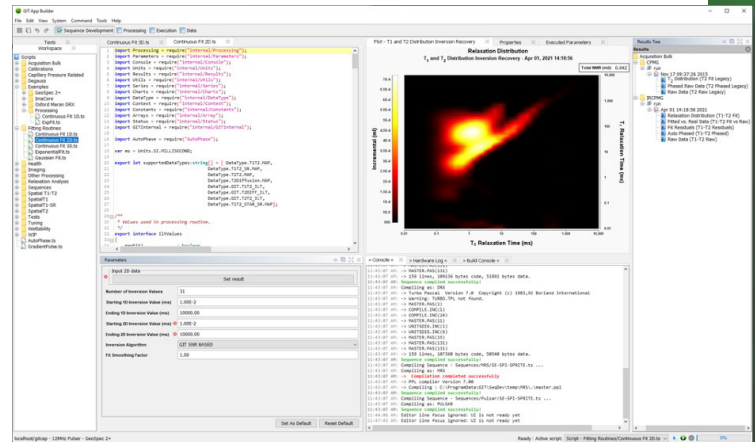


FIGURE 1 : Shows the development environment of NMR ProLab allowing users to configure internal functions and routine into user defined scripts to create turn-key processing scripts.

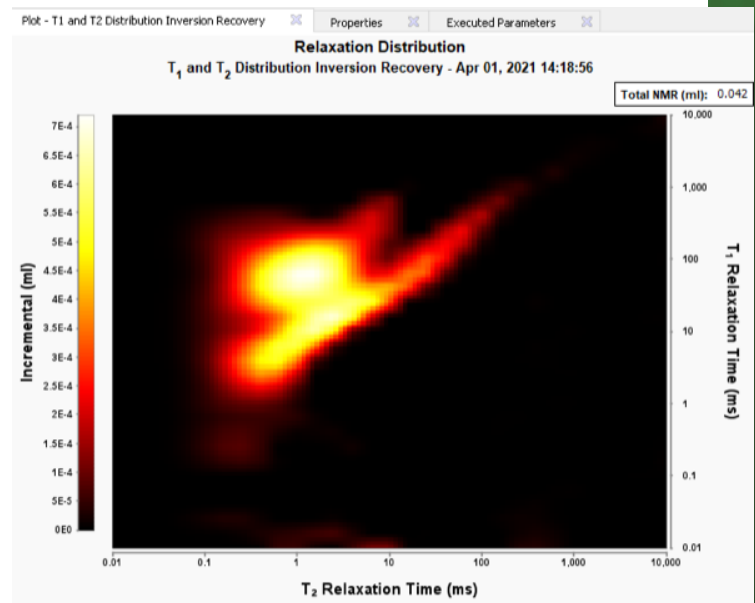


FIGURE 2 : An example of two-dimensional time domain NMR data processed into a 2D Map of continuously fitted relation times (T1 and T2 in this case).

Expand the information gained from your NMR data.

**NMRProLab** also provides a 3D rotational display for 3D processed data so users can interact with the data produced for further analysis or quality control.

By way of offering a level of processing automation, users can create user scripts using the built-in processing modules. This allows users to process large amounts of data with minimal user interaction.

GIT's **NMRProLab** can be used to process data from any time domain (TD) NMR instrument. Users can process data in the lab, in the office or when working remotely, as **NMRProLab** does not require direct access to the NMR instrument.

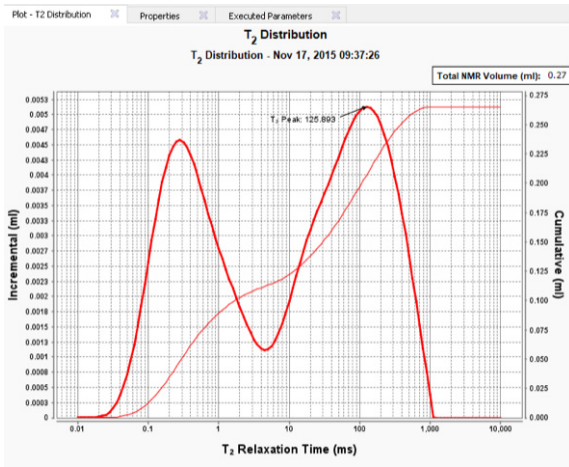


FIGURE 3 : An example where one-dimensional time domain NMR data is processed into a continuously fitted relaxation times (T2 in this case).

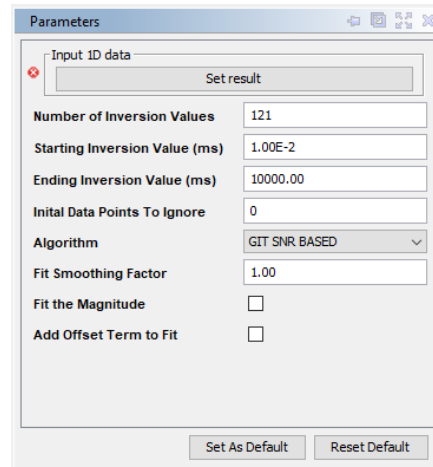


FIGURE 4: Easily adjust processing parameters to optimize your processing.

```

runs.ts Acq-DRX.ts Acq-Pulsar.ts Parameters.ts Continuous FR 2D.ts Acq-DRX.ts
73 inputResult : Parameters.ResultParameter,
74 isDiffusionScan : Parameters.Boolean
75 }
76 }
77 /**
78  * Main ILT 2D fitting function.
79  */
80 export function process(iltPars:ILTValues):Results.Result[]
81 {
82   Status.message("Started 2D Continuous Fitting");
83   Status.progress(0);
84
85   //Extract required parameters from iltPars and define defaults if not defined.
86   let kernelTypeID = Processing.KernelType.T2;
87   let kernelType2D = Processing.KernelType.T1;
88   let magFit = Utils.Optionals.orElse(iltPars.magFit, false);
89   let startBin1D = iltPars.startBin1D;
90   let endBin1D = iltPars.endBin1D;
91   let numBins1D = iltPars.numBins1D;
92   let startBin2D = Utils.Optionals.orElse(iltPars.startBin2D, iltPars.startBin1D);
93   let endBin2D = Utils.Optionals.orElse(iltPars.endBin2D, iltPars.endBin1D);
94   let numBins2D = Utils.Optionals.orElse(iltPars.numBins2D, iltPars.numBins1D);
95   let iltAlgorithm = Utils.Optionals.orElse(iltPars.inversionAlgorithm, Processing.InversionAlgorithm);
96   let alpha = Utils.Optionals.orElse(iltPars.alpha, 0.025);
97   let alphaFactor = Utils.Optionals.orElse(iltPars.alphaFactor, 1.0);
98   let addOffset = Utils.Optionals.orElse(iltPars.addOffset, false);
99   let dataType = iltPars.dataType;
100  let calibration = Utils.Optionals.orElse(iltPars.scalingFactor, 1);
101  let signal = iltPars.signal;
102  let noise = iltPars.noise;
103  let signalLocation = (iltPars.inputSeries.data.size(1) - 1) * iltPars.inputSeries.data.size(0);
104  let doFirstDimensionThenSecond = false;
105  let resultantDataType = DataType.T1T2_MAP;
    
```

FIGURE 5: Using TypeScript language easily access over 100 prebuild processing routines and design your own.

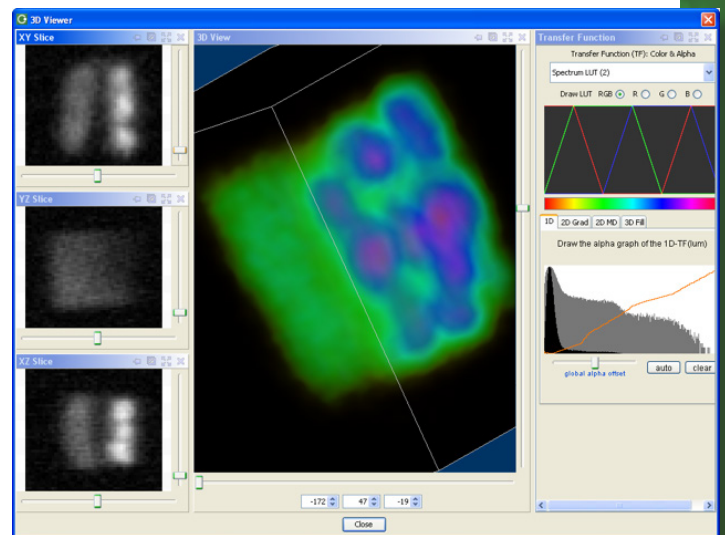


FIGURE 6: View 3D data with the include robust three-dimensional viewing tool.