WOLFRAM TECHNOLOGY CONFERENCE 2007

The Development of a Quality Control and Analysis Application for the ThermoFluor® High Throughput Screening Assay

Robert B. Nachbar[1]
Delphine Collin[2]
Jonathan Robinson[1]
Thomas J. Mildorf[3]
Eugen Buehler[1]

[1]Applied Computer Science and Mathematics
[2]Automated Biotechnology
[3]Massachusets Institute of Technology

Introduction

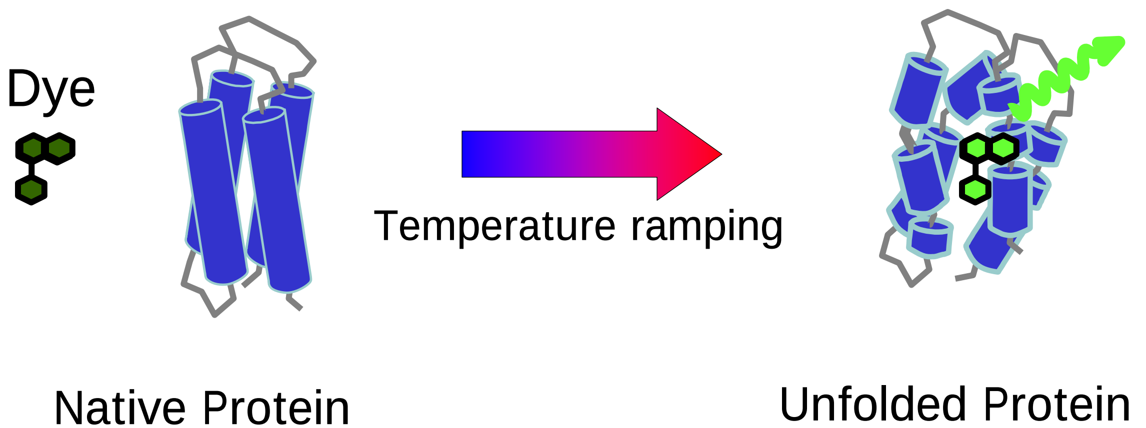
- Goal: Facilitate and log the work flow for a user analyzing the results from one high throughput screening (HTS) plate that employs the ThermoFluor® assay technology
- Problem: Analysis using vendor supplied software was time consuming and had low quality control due to reliance on user to grade the response of each well
- Solution: Create computerized tool to facilitate the analysis

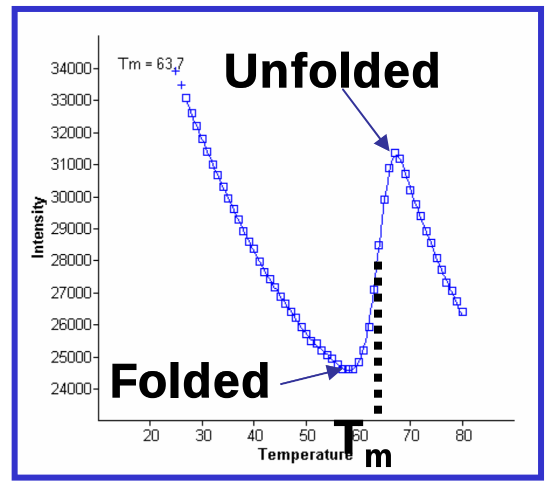
Background

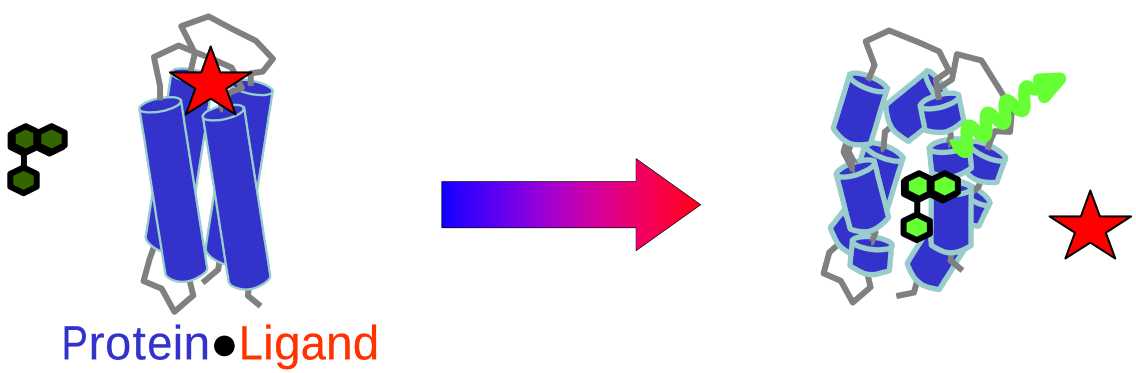
ThermoFluor®

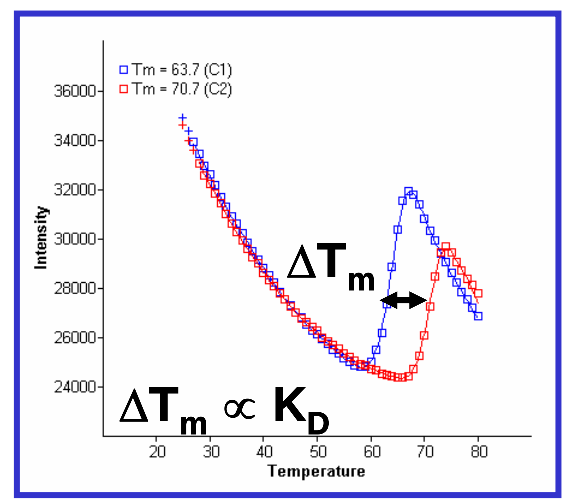
- Detection of unfolding of cytosolic proteins
- Characterization of specific binders
- Ranking as function of binding strength
- Throughput: 384 well plate, up to 7000 thermograms/24hrs
- High protein consumption: 1mg ~ 1500 samples

ThermoFluor®: Fluorescent Detection of Protein Unfolding









Developing the Requirements

Goals

Quality Control

- Obvious problems
  - spikes
  - high fluoresence
  - no obvious transition
- Distorted transition
- Large $|\Delta T_m|$
- Multiple transitions
- Check whole plate for unusual patterns
- User is final authority

Analysis

- Smooth thermogram
- Calculate gradient, curvature, and critical points
- Calculate T_m for each well
- Calculate mean T_m for control wells
- Calculate ΔT_m for sample wells
- Export results to Excel file for data repository
- Audit trail notebook

Developing the Tools

Import Data from Robot

Use built-in `Import` function

```
In[1]:= TFD =
  Import["C:\\nachbarr\\Documents\\Software\\Wolfram Research
    \\src\\HTS\\testing\\      .txt", "ThermoFluor"]

Out[1]= ThermoFluorData[<16, 24>, ControlWells → <32>, SampleWells → <352>
  , EmptyWells → {}, ValidWells → {}, InvalidWells → {}]
```

`ThermoFluorData` object returned

Special Format defined that uses skeletons

Smooth Data, Calculate Critical Points and Transitions

```
In[2]:= TFD = FindCriticalPoints[TFD, Smoothing → {"SavitzkyGolay[3]", 5},
          TemperatureRange -> {25, 75}]

Out[2]= ThermoFluorData[<16, 24>, ControlWells → <32>
        , SampleWells → <352>, EmptyWells → {},
        TemperatureRange → {25, 75}, Smoothing → {SavitzkyGolay[3], 5}
        , SmoothData → {<51>, <384, 51>, <384, 51>},
        CriticalPoints → <384>, ValidWells → {}, InvalidWells → {}]
```

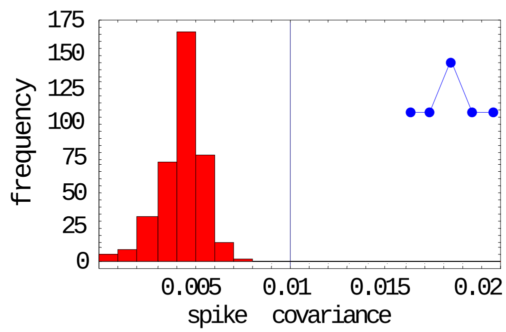
User's options and new results are appended

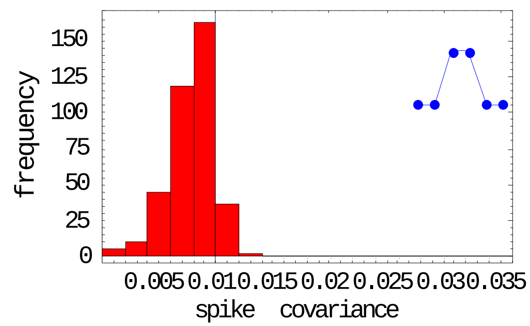
Running list of valid & invalid wells maintained

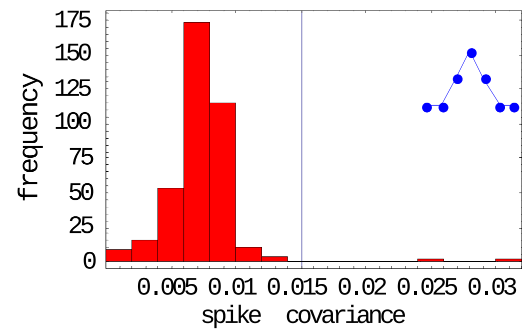
```
In[3]:= TFD = FindTransitions[TFD, PseudoMaxMinRelativeGradient → 0.5,
          TmMinimumRelativeIntensity → 0.05, TmMinimumRelativeGradient → 0.25]

Out[3]= ThermoFluorData[<16, 24>, ControlWells → <32>
        , SampleWells → <352>, EmptyWells → {},
        TemperatureRange → {25, 75}, Smoothing → {SavitzkyGolay[3], 5},
        SmoothData → {<51>, <384, 51>, <384, 51>}, CriticalPoints → <384>
        , Transitions → <384>, PseudoMaxMinRelativeGradient → 0.5,
        TmMinimumRelativeIntensity → 0.05, TmMinimumRelativeIntensity → 0.25
        , NoTmFailed → <11>, ValidWells → <373>, InvalidWells → <11>]
```

Spike Filter → Glitch Filter

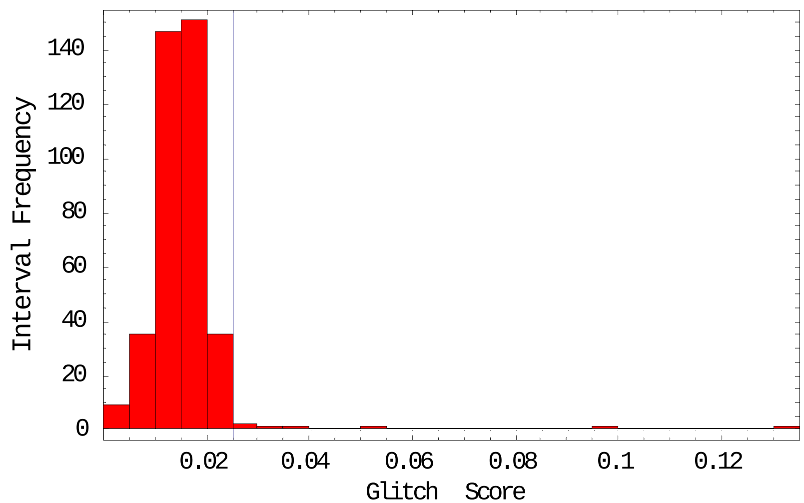
In[8]:= **FilterSpikes[TFD, Thresholds → Automatic]**

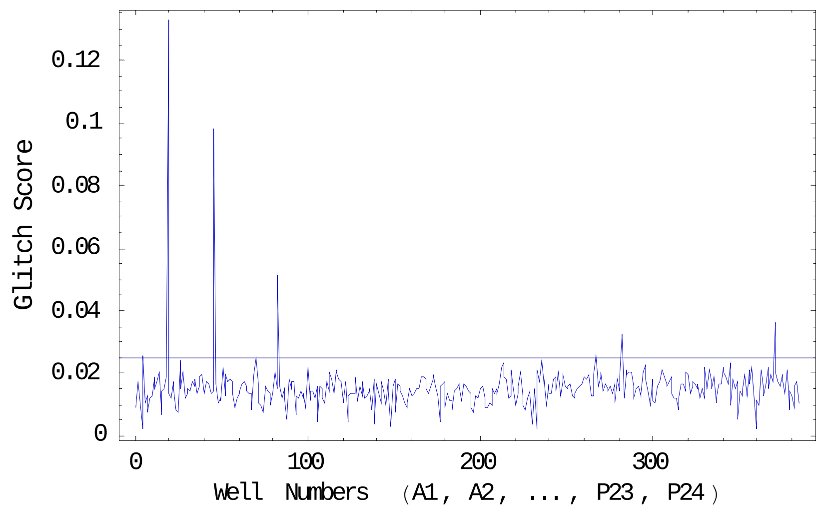




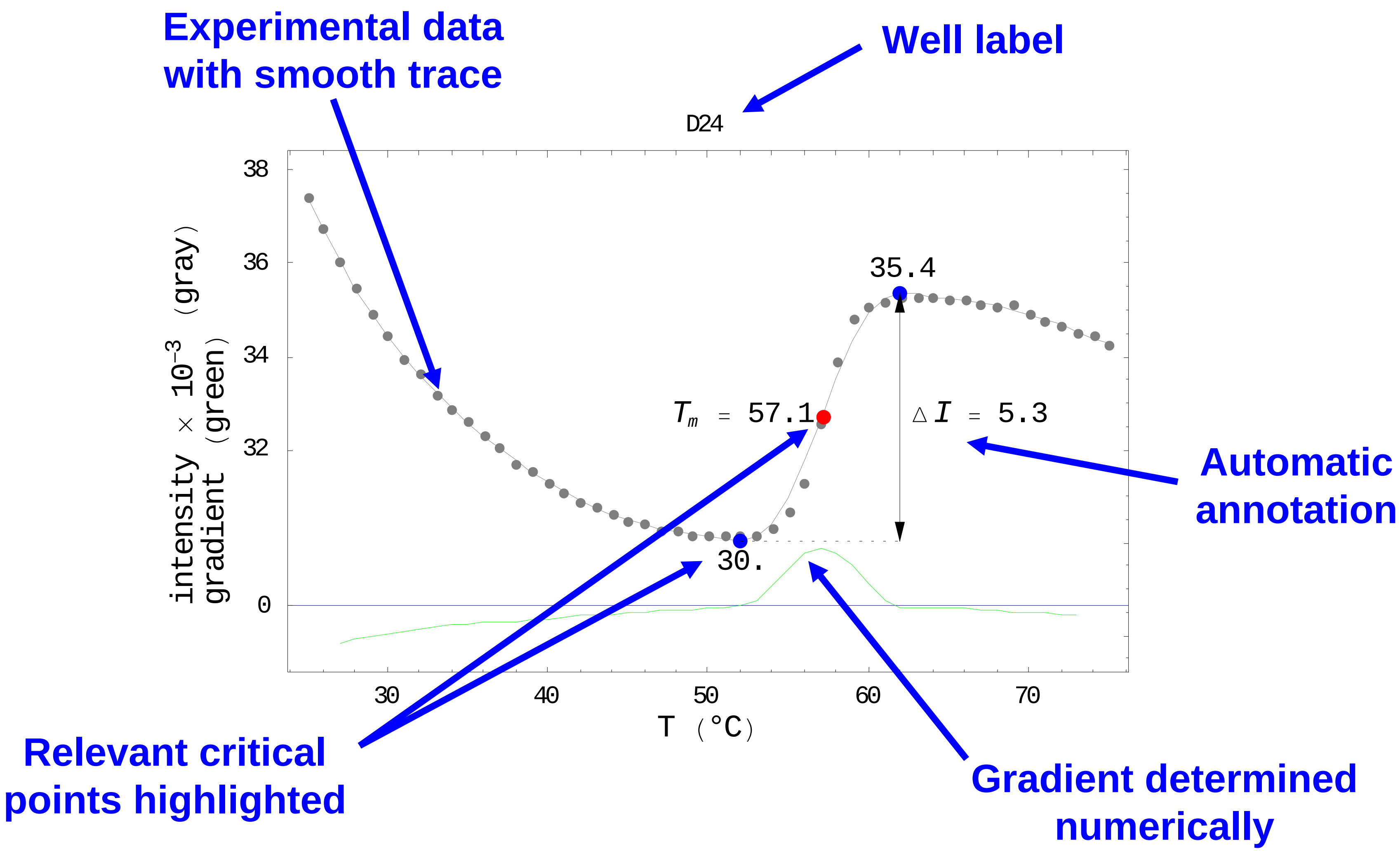


FilterGlitches[TFD, Thresholds → Automatic]

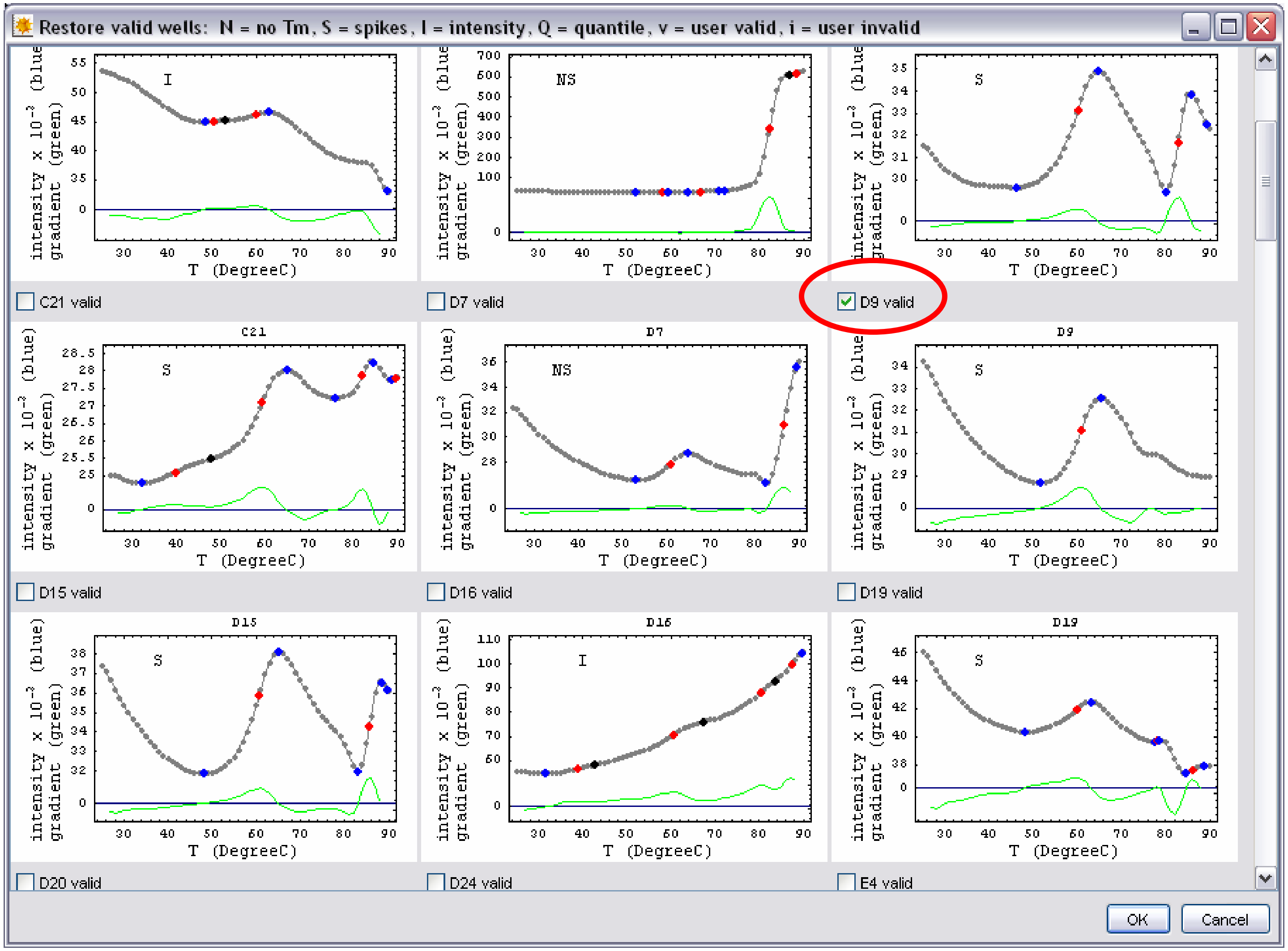




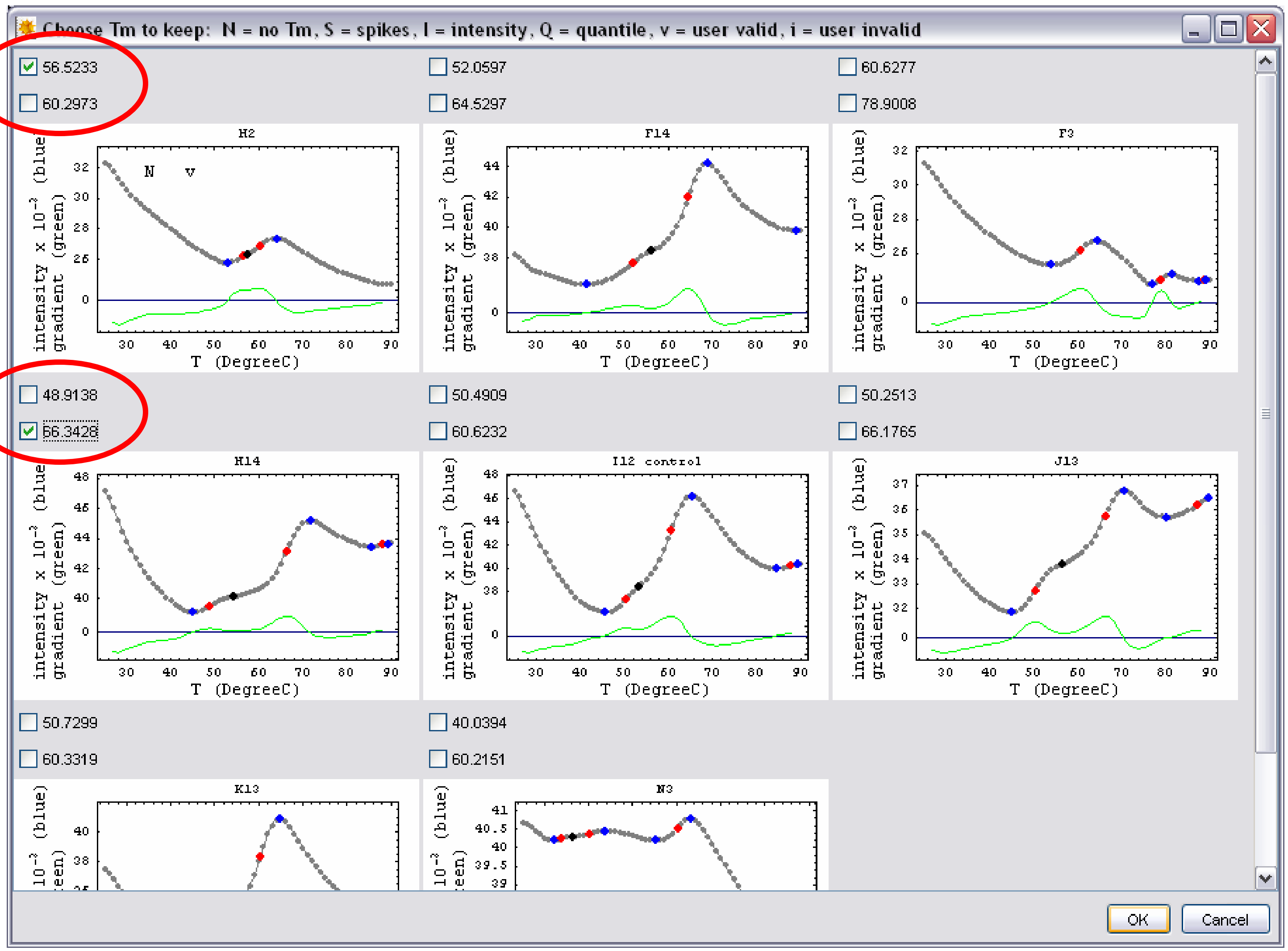
Thermogram display developed for the user

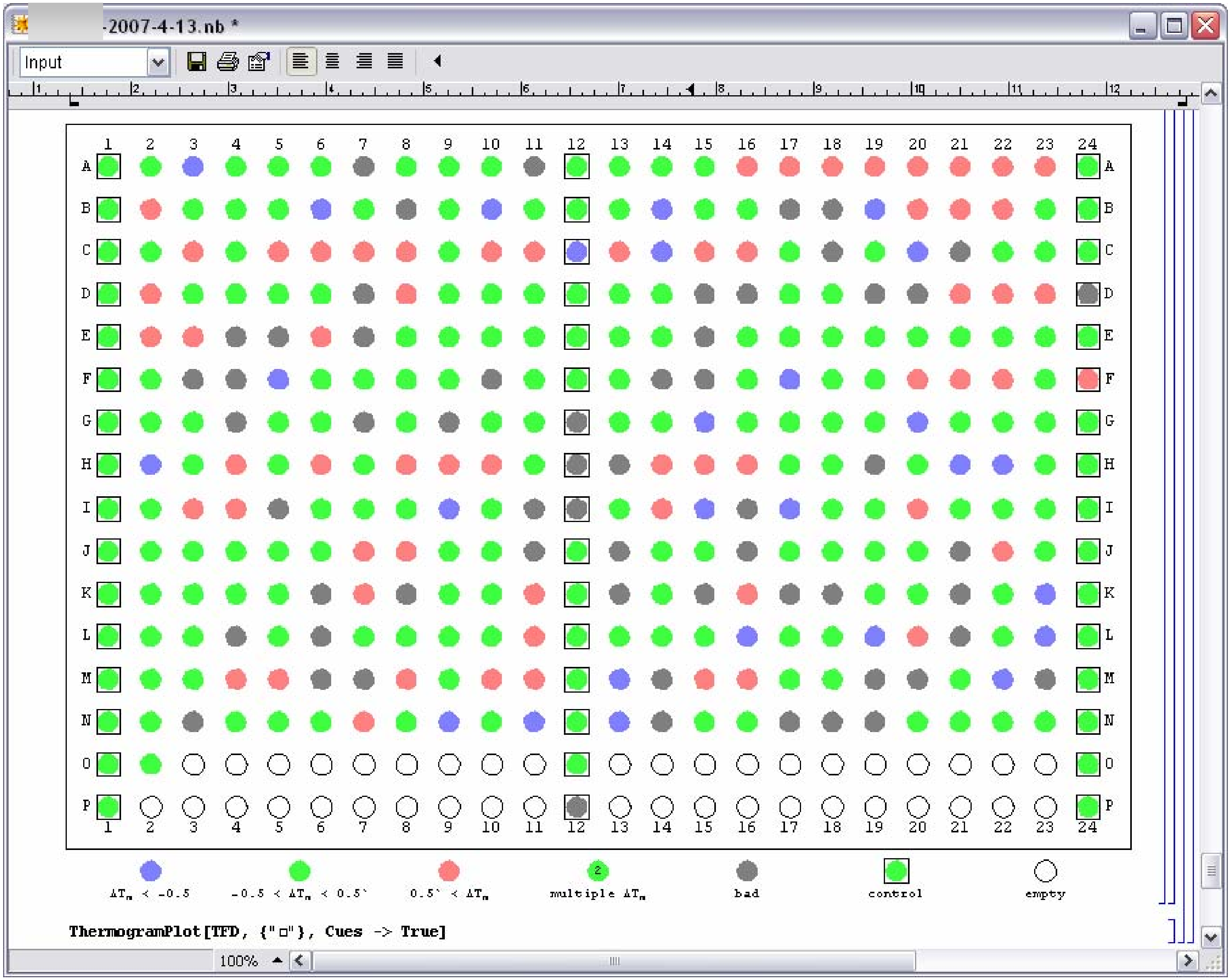


User Can Override Automated QC



User Can Choose Among T_m




-2007-4-13.nb *
Input
ΔT_m < -0.5
-0.5 < ΔT_m < 0.5`
0.5` < ΔT_m
multiple ΔT_m
bad
control
empty
ThermogramPlot[TFD, {"□"}, Cues -> True]
100%

Lessons

- Statistics could not be used to set thresholds—the errors are not normally distributed.
- Principal components and clustering could not be used to find outliers—too many false positives and false negatives.
- Machine learning classifiers were not successful.
- Heuristics and interactive user input worked.
  - The challenge was to make it efficient!
- Frequent dialog with the users identified simple improvements that made big differences.
- Experimental background of developer facilitated communication with users.

Developing the Application

Original Version of ThermoFluor Analysis in *Mathematica* Notebook

- First version created in *Mathematica* allowed for quantitative analysis of ThermoFluor plates
- Users needed to know *Mathematica* syntax and enter specific commands manually
- A GUI to guide the user is much easier to use

```
ToFileName[{$HomeDirectory}] // SetDirectory;
```

```
ToFileName[{$SourceDirectory, "HTS"}, "ThermoFluor.m"] // Get
```

```
plateData = Import["../      .txt", "ThermoFluor"]

-ThermoFluorData[<16, 24>, ControlWells → <48>, SampleWells → <295>
 , EmptyWells → <41>, ValidWells → <384>, InvalidWells → {}]-
```

```
SetOptions[FindCriticalPoints, Smoothing → 3, Window → 2]
SetOptions[FindTransitions, TmRelativeIntensity → 0.05]

{Smoothing → 3, Window → 2}
```

```
{TmRelativeIntensity → 0.05}
```

Outsourcing

- Decision to outsource interface development allowed for internal focus on development and refinement of algorithms for ThermoFluor HTS plate analysis.
- Wolfram's Accredited *Mathematica* Consultants and Consulting Companies helped find a qualified consultant.
- Formal requirements document allowed consultant to easily scope the work and estimate cost.
- Punch list after delivery and user testing helped resolve all the issues.
- Outsourcing user documentation is not easy.

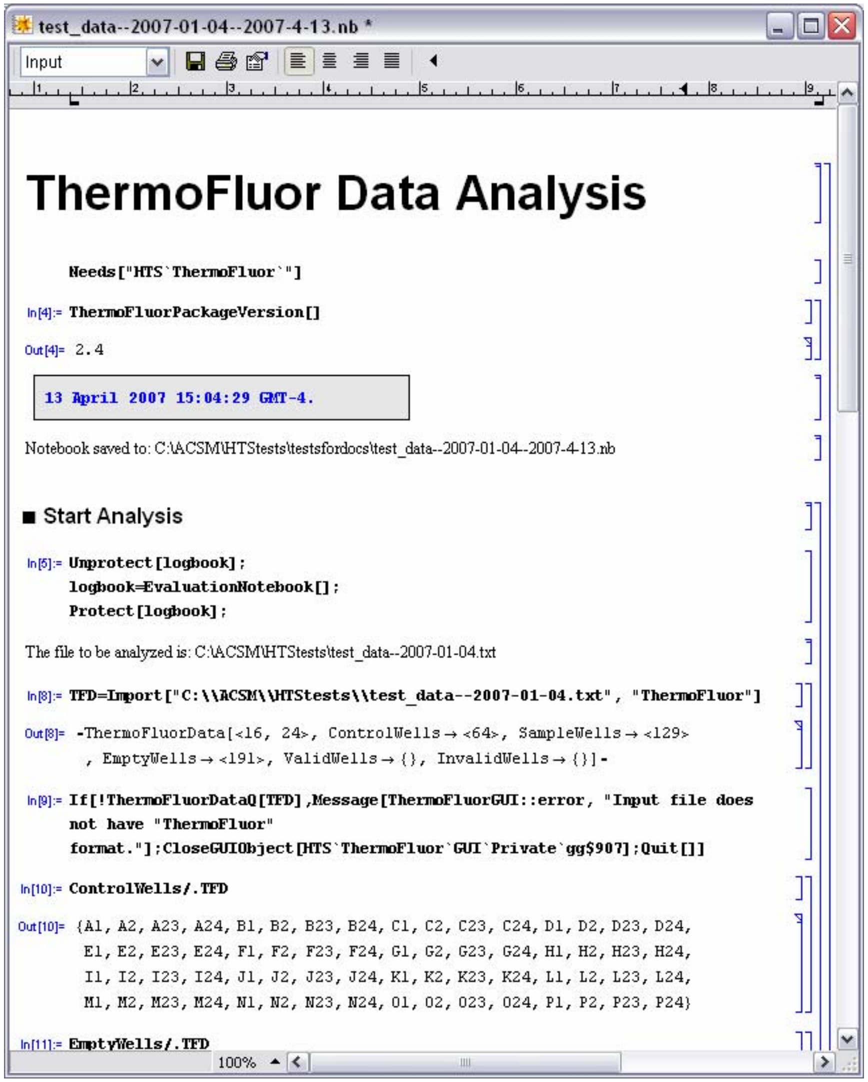
New Version

- Graphical interface was outsourced
  - Included development of menu to guide users through workflow and some additional windows
  - Graphical interface would allow for much quicker adoption of tool by making it user friendly and reducing training time
- Menu guides users through workflow

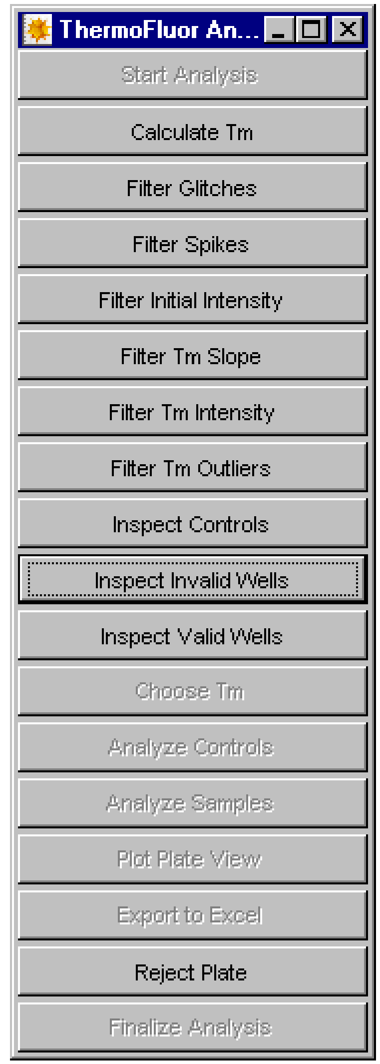
Live Demonstration

continue

Workflow Excerpts



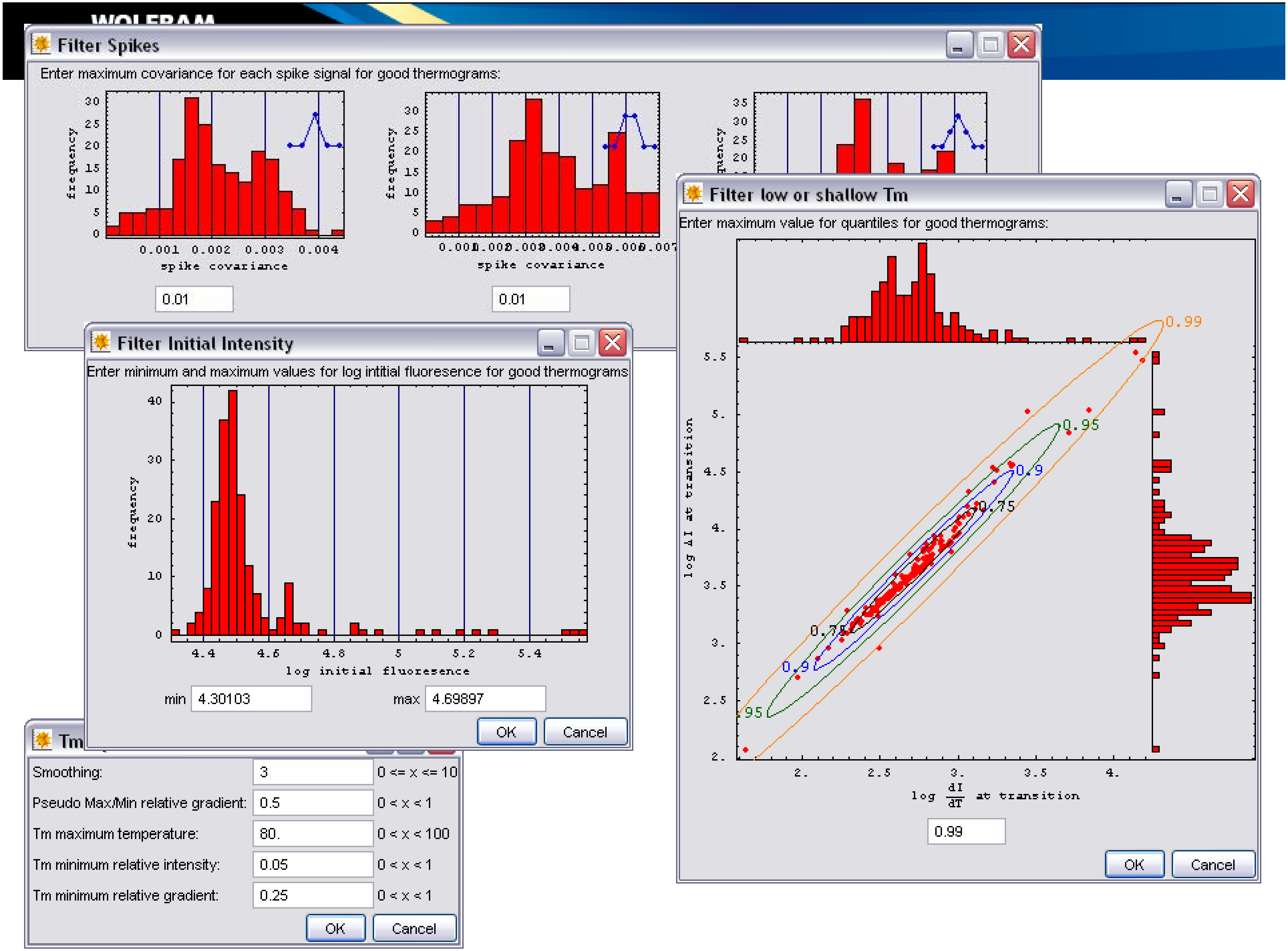
Audit trail notebook



Buttons for valid next actions are active

Button for logical next action is selected

Palette to guide user


Filter Spikes
Enter maximum covariance for each spike signal for good thermograms:
frequency
spike covariance
0.01
0.01
Filter Initial Intensity
Enter minimum and maximum values for log intitial fluoresence for good thermograms
frequency
log initial fluoresence
min 4.30103
max 4.69897
OK
Cancel
Filter low or shallow Tm
Enter maximum value for quantiles for good thermograms:
0.99
0.95
0.9
0.75
log ΔI at transition
log dI/dT at transition
0.99
OK
Cancel
Smoothing: 3 0 <= x <= 10
Pseudo Max/Min relative gradient: 0.5 0 < x < 1
Tm maximum temperature: 80. 0 < x < 100
Tm minimum relative intensity: 0.05 0 < x < 1
Tm minimum relative gradient: 0.25 0 < x < 1
OK
Cancel

Lessons

- Reuse of high-level graphics functions in GUI components speeds development.
- Notebook programming for the audit trail is not difficult.
- Some tinkering is needed to get GUI components to appear on user's display in useful locations.
- *GUIKit* had all the components needed, except one—a customized file name *and* file type dialog was written in Java.
- Evaluations in *GUIKit* palette and audit trail notebook are asynchronous, which permits the user to get ahead of her/himself.
- Logic for workflow control with *GUIKit* is missing some critical pieces.

Conclusions

Benefits to the User

- Time Reduction for Analysis
  - Previous analysis by hand for one 384 well HTS plate took 2 days
  - Using the new software, analysis of a plate can be completed in 30 minutes or less
  - On a 30 plate assay, the estimated time savings is 12 person weeks
- More Consistent Analysis
  - Use of software produces more consistent analysis across all the plates

Acknowledgements

- ScienceOps