What's New Pulseq

Maxim Zaitsev

Division of Medical Physics, Dept. of Radiology,
University Medical Center Freiburg, Germany

Nov 16, 2023

What is 'New' in a Continuum?

- Some interesting/relevant developments from the past ~12 months

- RF simulation
- Peripheral nerve stimulation (PNS)
- Data labels, slice positions
- Mechanical resonances

- Some more fun stuff…

RF Simulation & Analysis

In addition to TOPPE's slice profile calculation…

mr.calcRfCenter() & **mr.calcRfBandwidth**()
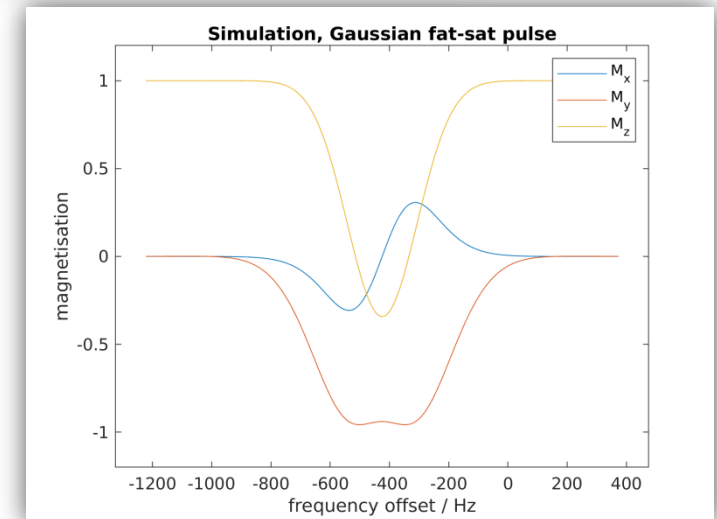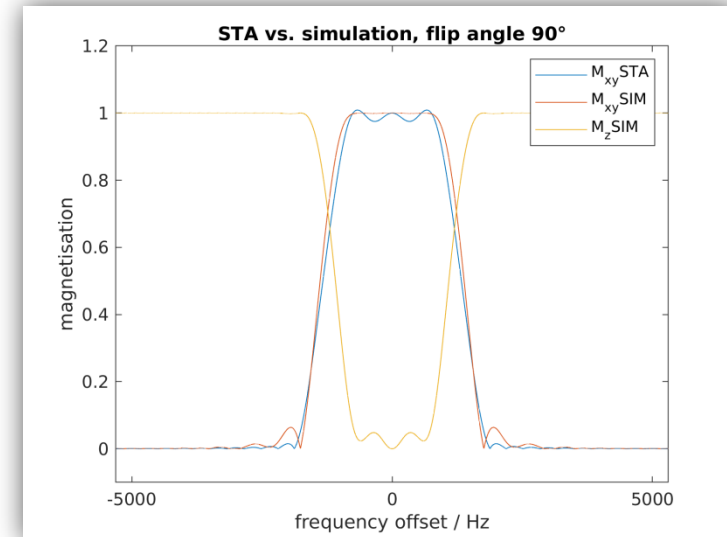
- return RF peak position / FT width

- Fast, only for approximate guidance,
  small tip-angle approximation (STA)

[Mz_z,Mz_xy,F,ref_eff,Mx_xy,My_xy]=

       mr.simRf(rfPulse, preph_fact, reph_fac)

- Fast RF simulation based on the rotation formalism
  (ignoring relaxation)

See demoUnsorted/demoRfSimulation.m in the main
Pulseq repository for numerous use examples

Peripheral Nerve Stimulation Prediction

- Direct integration with the SAFE model (Siemens-specific)
- Open-SAFE by Filip Szczepankiewicz
  https://github.com/filip-szczepankiewicz/safe_pns_prediction

seq.calcPNS()

- Analyze the sequence in memory for PNS

- Ideally requires access to the pertinent scanner-specific PNS model data (stored in *.asc files)
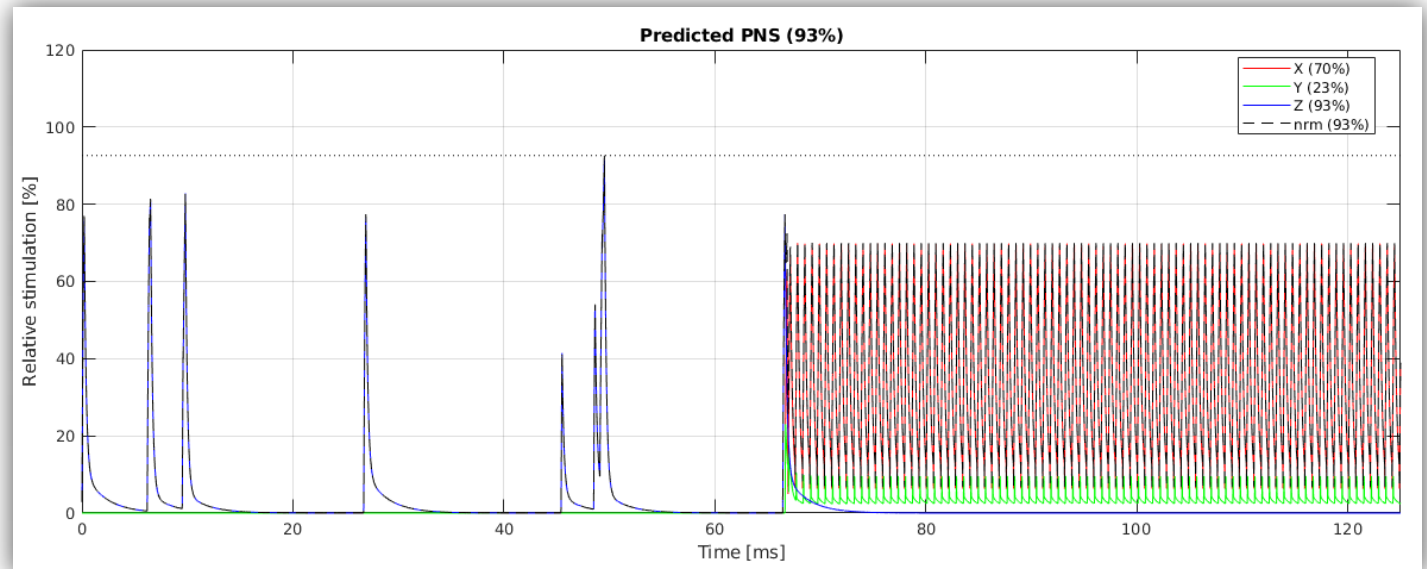
Pulseq Extensions

- Pulseq file format readily supports very flexible Extension objects
- Established extensions
  - Cardiac triggering
  - Trigger (digital output) pulses
  - Data labeling (the traditional use of the LABEL extension)
- Labels to control sequence execution
  - ONCE labels
  - NOPOS, NOROT, NOSCALE labels
- Labels may provide hints for the interpreters
  - New GE (and possibly candidate Philips) interpreters detect the "TR Loop"
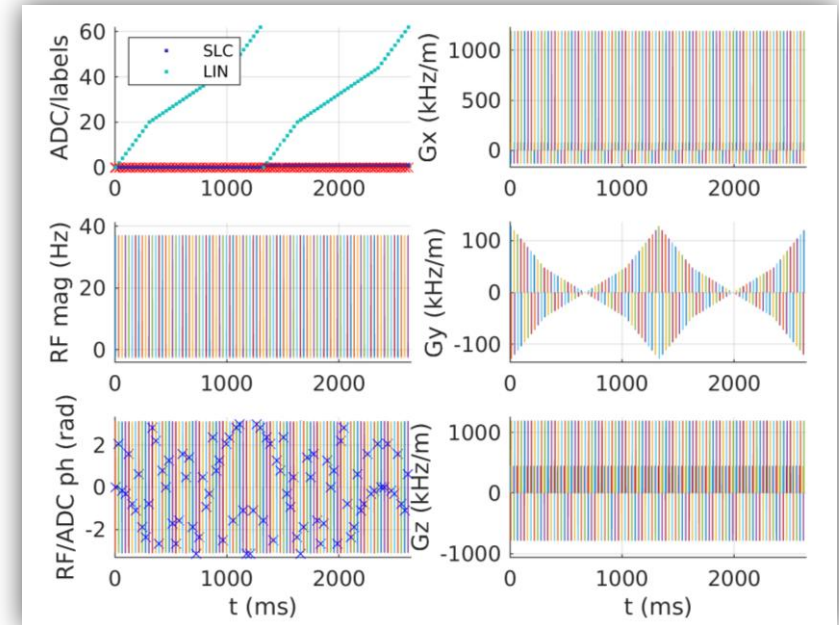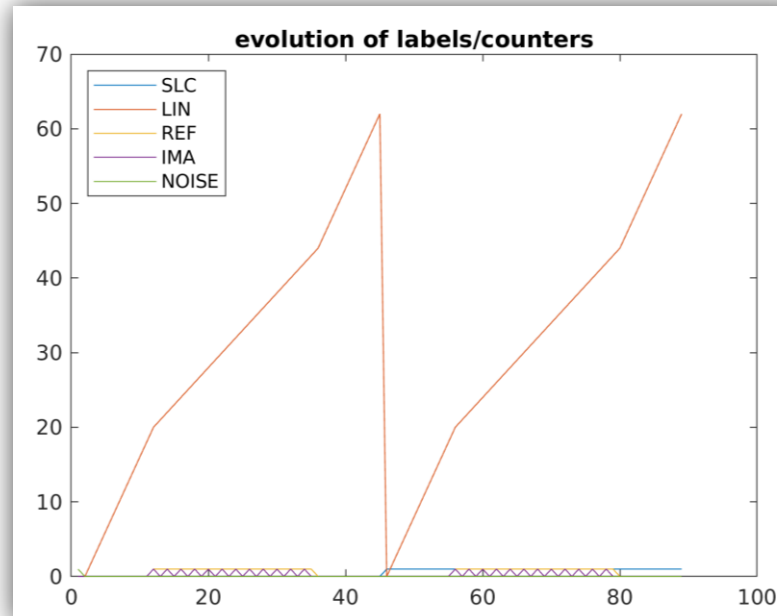
New Label Display and Debugging Options

- Labeling of the raw data is often desirable
  - Online recon & integration with Gadgetron
- seq.plot('TimeDisp', 'ms', 'label', 'lin,slc');
- adc_lbl=seq.evalLabels('evolution','adc');

```
adc_lbl =

  struct with fields:

      LIN: [0 0 2 4 6 8 10 12 14 16 18 20 ... 37 38 39 40 41 42 43 44 46 48 50 52 54 56 58 60 ... ]
      IMA: [0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 ...      0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 ... ]
      REF: [0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 ...      0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ... ]
    NOISE: [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ... ]
      SLC: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...      0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ... ]
```

New Labels: ONCE

- Some interpreters offer the option run/repeat the Pulseq sequence multiple times without a delay

- Label ONCE when set to

1. Mark blocks that are only executed <u>on the first</u> repetition

2. Mark blocks that are only executed <u>on the last</u> repetition

- Possible use cases:
  - Calibration scans performed once at the beginning
  - Signal preparation
    - dummy scans in GRE or
    - α/2 TR/2 preparation in trueFISP

- As example see <u>demoSeq/writeTrufi.m</u>
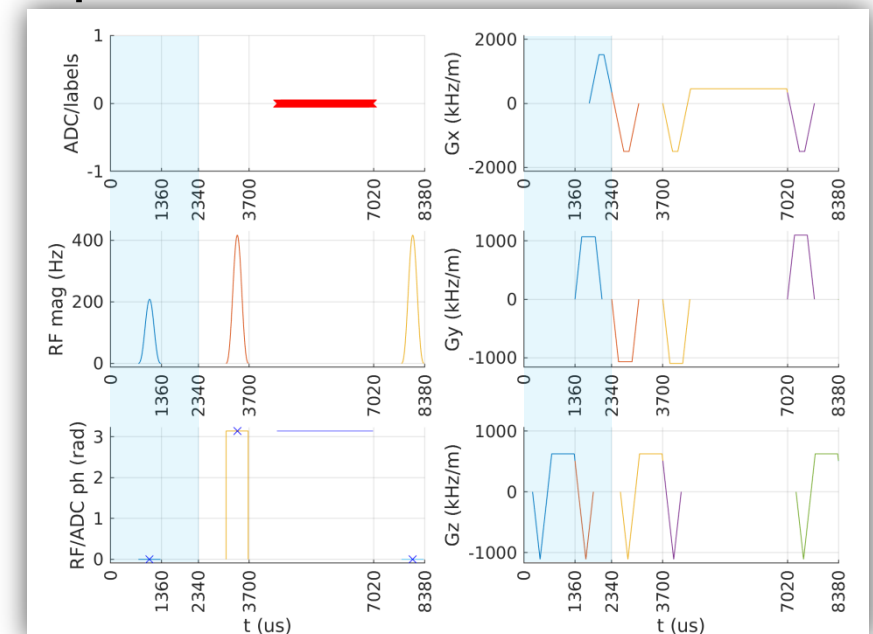
Multi-Slice Mode

- Traditionally Pulseq presents itself as a "3D sequence"
    - FOV in the [Definitions] section inscribes all slices
    - FOV positioning works similarly for all sequences
    - Positions of individual slices are not visible to the user

- New definitions switch the sequence to a 2D multi-slice mode:
    seq.setDefinition('SlicePositions', slicePositionsArray);
    seq.setDefinition('SliceThickness', thickness_in_m);
    seq.setDefinition('SliceGap', sliceGap_in_m);

3D-mode vs. 2D-mode
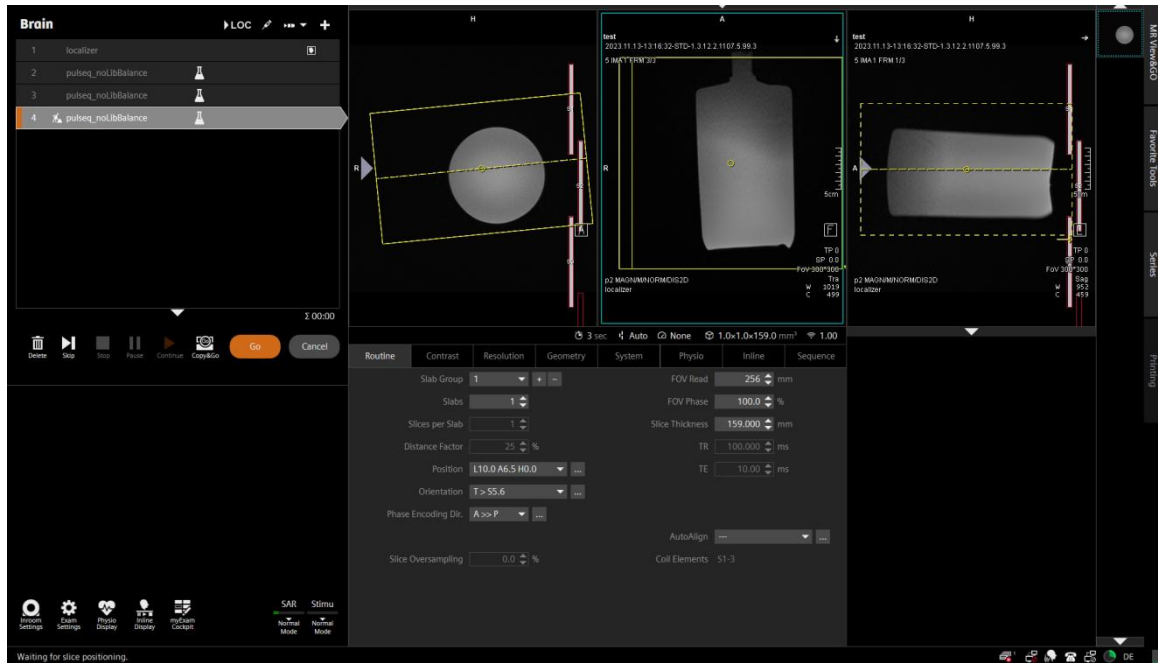
'legacy' 3D mode
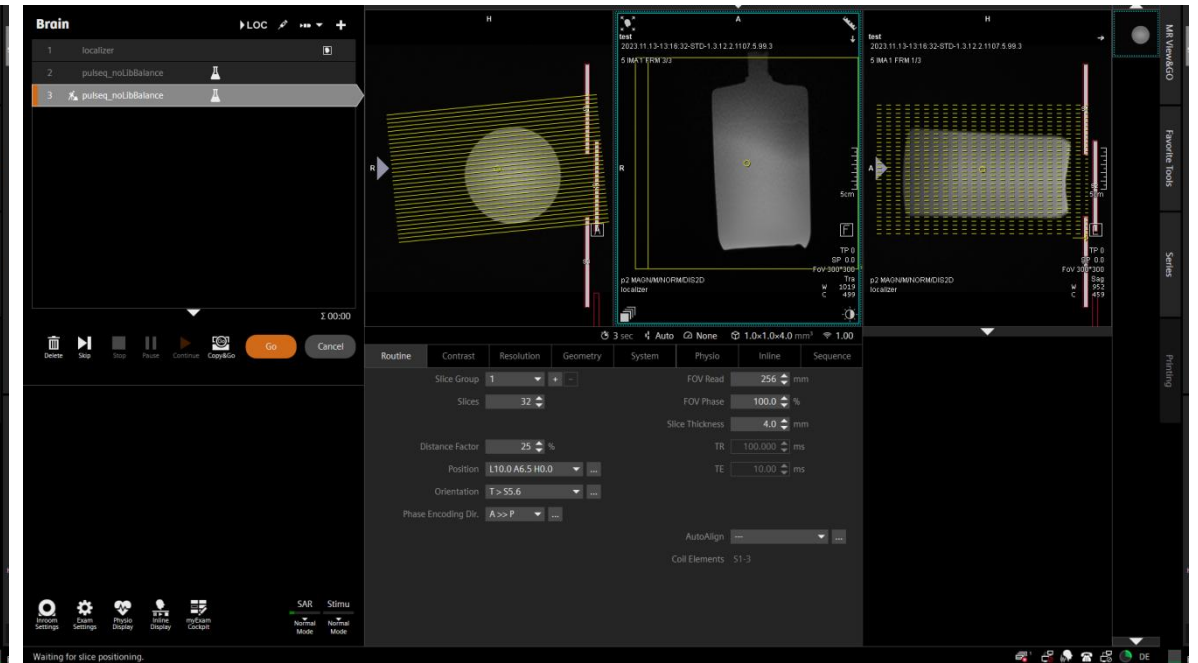
new 2D multi-slice mode

- Homogeneous handling for all imaging and spectroscopy sequences

- Inner life of the sequence is opaque

- 2D sequences need special handling

- <u>Correct slice position information</u> in raw data

- More familiar user experience

Gradient Spectrogram Analysis

- CAUTION! Gradient coil have **mechanical resonances** (forbidden echo spacings)

- Pulseq now provides a demo (not a function yet) for a spectral analysis of the sequence in memory, see demoUnsorted/gradSpectrum.m

- We would like to hear from the manufacturers how to handle this knowledge

UNIVERSITÄTS
KLINIKUM FREIBURG
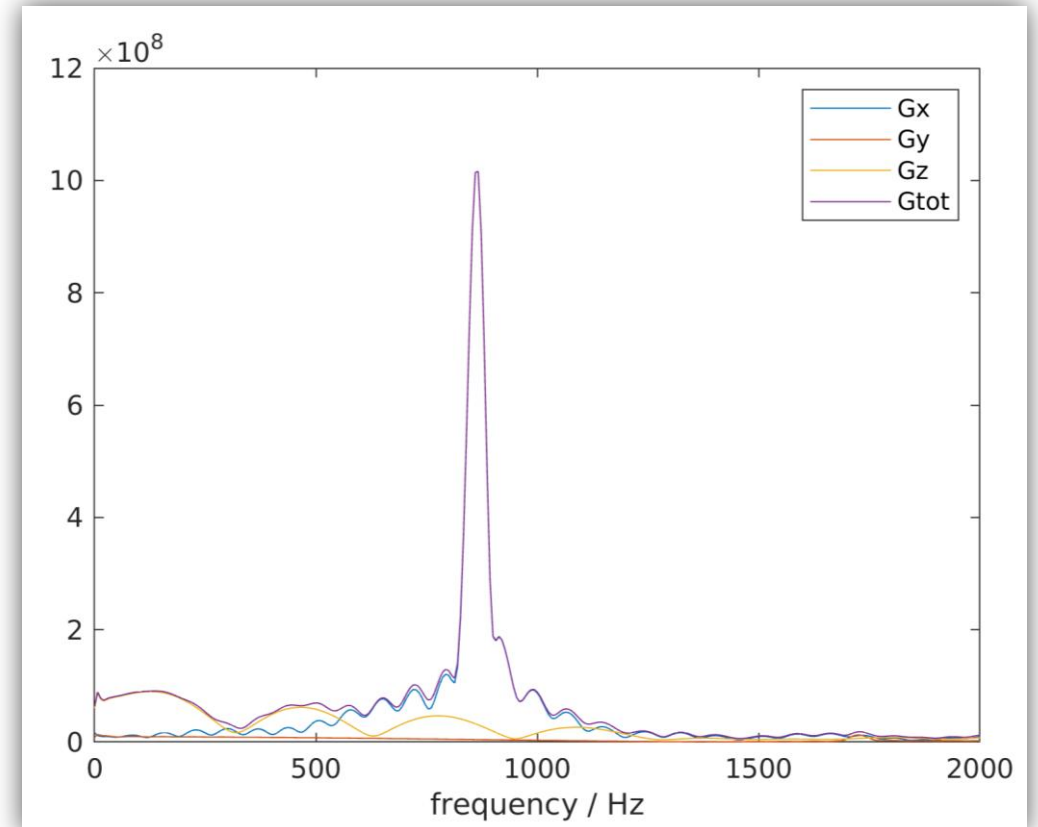
Ready for the Fun Part?

Pulseq & Gradient Music

- Experimental **mrMusic** library for creating gradient music

DISCLAIMERS:

- **You** are **completely, entirely and absolutely** responsible for the health or damage of your gradient system
- **Always** check the gradient spectrum and **avoid** the **mechanical resonances** of your system
- **Never** use the **maximum** gradient **amplitude**

mrMusic how-to

- Located in demoUnsorted of the main repository (lib + 4 demo scripts)
  - musicBadinerie.m, musicEntertainer.m, musicRootBeerRag.m & musicSoundTest.m

1. Initialize the mrMusic environment:
   ```
   mrMusic.init; % this creates over 30 global variables
   ```

2. Create the melody as a cell array of bars and channels (n x 3)
   ```
   melody = { [ … ],[ … ],[ … ];}
   ```

3. Convert melody to a channel-frequency table
   ```
   [pitches, durations] = mrMusic.melodyToPitchesAndDurations(…);
   ```

4. Populate the Sequence object with the music piece
   ```
   seq = mrMusic.musicToSequence(seq, …);
   ```

5. Check how it sounds
   ```
   seq.sound();
   ```

More about Melody and Notes

```
melody = {
    % bar 1
    [ o/4 h2/8  d3/16 h2/16 ],...
    [ o/4 d2/8  o/8          ],...
    [ o/4 hbb/8 hbb/8        ];...
    % bar 2
    [ fis2/8 h2/16 fis2/16 d2/8   fis2/16 d2/16 ], ...
    [ h1/8   o/8                fis1/8 o/8           ], ...
    [ db/8   hbb/8              fisb/8 db/8          ]; }
```

- Notes: a, b, c, d, e, f, g, h; '_is' is high '_es' is flat; 'a1' is the middle A
- Octaves: _bb, _b, _1, _2, _3;
- 'o' is a delay
- To define duration divide the note by the fractional duration, e.g. cis2/16 is C# of the second octave with the duration of 1/16th
- **Enjoy**!

mrMusic: Concluding Remarks

- Really easy to create nice 3-voice tunes
- Music can be combined with imaging or spectroscopy
- Many more options available (explore examples)
  - Sound quality settings (wave vs. sawtooth)
  - Temperaments for Baroque music
  - Staccato & Legato
  - **Tuning for avoiding mechanical resonances**
  - …

- **Enjoy responsibly!**

Acknowledgements:

Berkin Bilgic	Borjan Gagoski	Douglas Noll
Frank Zijlstra	Imam Shaik	Jeff Fessler
Jon-Fredrik Nielsen	Juergen Hennig	Mojtaba Shafiekhani
Moritz Zaiss	Naveen Murthy	Niklas Wehkamp
Qiang Liu	Qingping Chen	Scott Peltier
Sebastian Littin	Will Grissom	Yogesh Rathi

THANK YOU FOR YOUR ATTENTION!