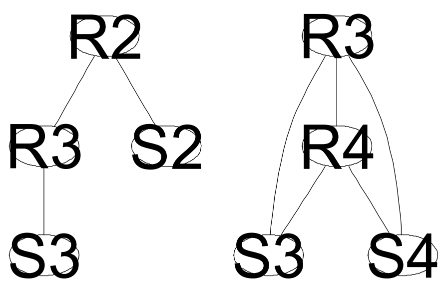
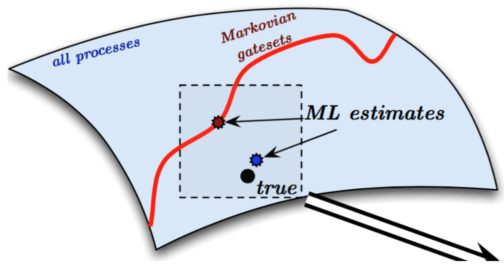
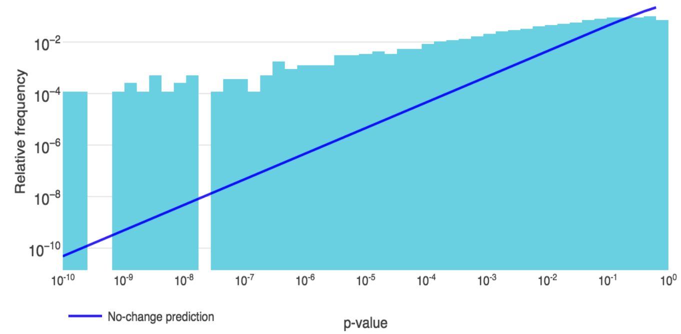
SAND2017-10601PE




# Sandia theory updates

*detecting crosstalk, drift, context dependence*

*pyGSTi updates*

Robin Blume-Kohout, Erik Nielsen,
Tim Proctor, Kenny Rudinger, Kevin Young

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525

# Outline

- Detecting differences between datasets
  - Depending on experiment design, can detect context dependence, drift, crosstalk
  - Expects chunked data
- Detecting crosstalk
  - Testing for conditional independence
- Detecting and characterizing drift
  - Time dependent changes in the measurement probabilities
  - Requires measurement record (not just total counts)
- Updates to pyGSTi
  - HTML reports, non-Markovian error bars, standard practice GST

# Detecting drift and crosstalk

- Do my qubit operations change when I change an “external variable?
  - E.g., today vs. tomorrow (drift), simultaneously operating on another qubit or not (operation crosstalk), etc.
- Can detect change by looking at differences in observed counts for same gate sequences.
- This framework:
  - Is sensitive to drift on timescales *greater* than individual sequence runtime.
  - Can use GST sequences but does not require GST analysis.
  - Is entirely agnostic to structure of underlying dynamics.
  - Relatively coarse-grained.

# Checking the equality of coin flips

| $h_A$ heads<br>$N_A$ flips | $h_B$ heads<br>$N_B$ flips |
| --- | --- |
| Data $A$ | Data $B$ |

# Checking the equality of coin flips

| | |
|---|---|
| $h_A$ heads<br>$N_A$ flips | $h_B$ heads<br>$N_B$ flips |
| Data *A* | Data *B* |

| **Crosstalk** | |
|---|---|
| Driven neighbor | Idle neighbor |
| **Drift** | |
| Today | Tomorrow |
| **Context** | |
| With fiber noise cancelation | Without fiber noise cancelation |
| With filter A | With filter B |

# Checking the equality of coin flips

| | |
|---|---|
| $h_A$ | heads |
| $N_A$ | flips |

Data $A$

| | |
|---|---|
| $h_B$ | heads |
| $N_B$ | flips |

Data $B$

| | |
|---|---|
| $h_A + h_B$ | heads |
| $N_A + N_B$ | flips |

Combined data $A + B$

# Checking the equality of coin flips

$p_A = h_A / N_A$

Data *A*

$p_B = h_B / N_B$

Data *B*

$p_{A+B} = ( h_A + h_B) / (N_A + N_B)$

Combined data *A* + *B*

**Hypothesis testing**

$H_0$: All data is drawn from the binomial distribution with $p_{A+B}$

$H_1$: Data *A* drawn from binomial distribution with $p_A$ and
Data B drawn from binomial distribution with $p_B$

# Checking the equality of coin flips

$p_A = h_A / N_A$

Data *A*

$p_B = h_B / N_B$

Data *B*

$p_{A+B} = (h_A + h_B) / (N_A + N_B)$

Combined data *A* + *B*

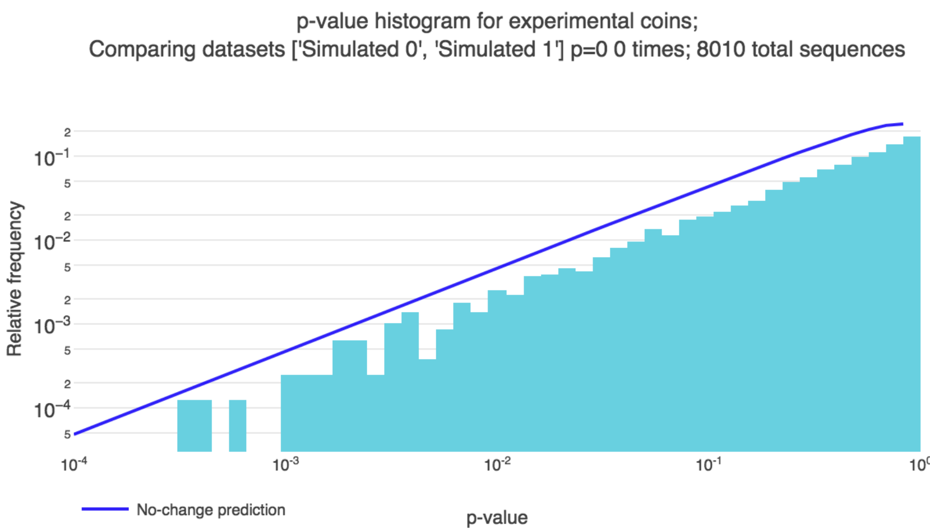
**Likelihood Ratio Test**

$$\text{Score} = -2\log\frac{\mathcal{L}_{A+B}}{\mathcal{L}_A\mathcal{L}_B} = -2\log\frac{p_{A+B}^{h_A+h_B}(1-p_{A+B})^{N_A+N_B-h_A-h_B}}{\left(p_A^{h_A}(1-p_A)^{N_A-h_A}\right)\left(p_B^{h_B}(1-p_B)^{N_B-h_B}\right)}$$

*If the distributions are the same, the score will be chi$^2$ distributed.*

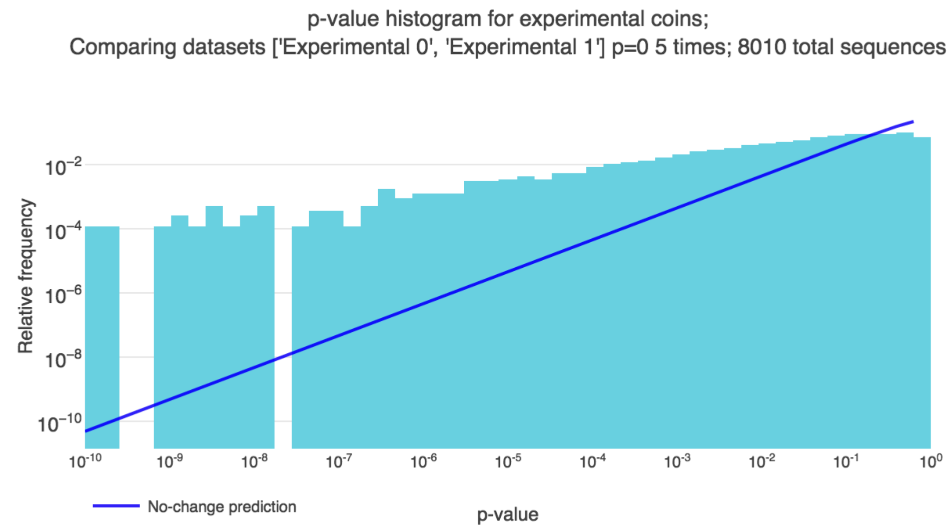
# Drift detection in Rigetti device

## Simulated Markovian GST data




Simulated data where the bias *does **not** change* across sweeps
(8010 gate strings)

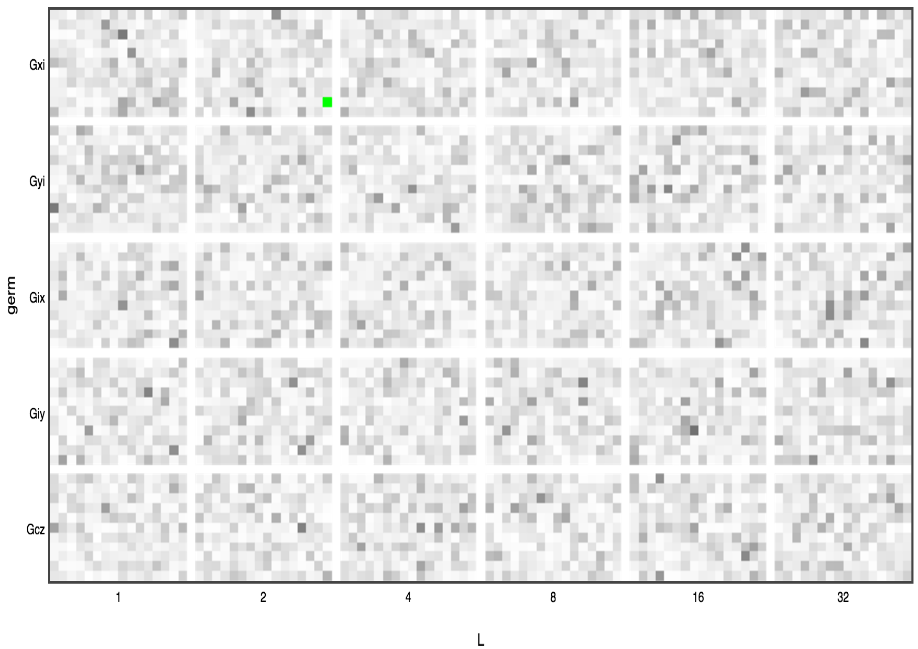
## Experimental GST data (Rigetti)




Experimental data where bias apparently ***does** change* across sweeps
(8010 gate strings).
“1 in ∞” chance the gate sets are the same.

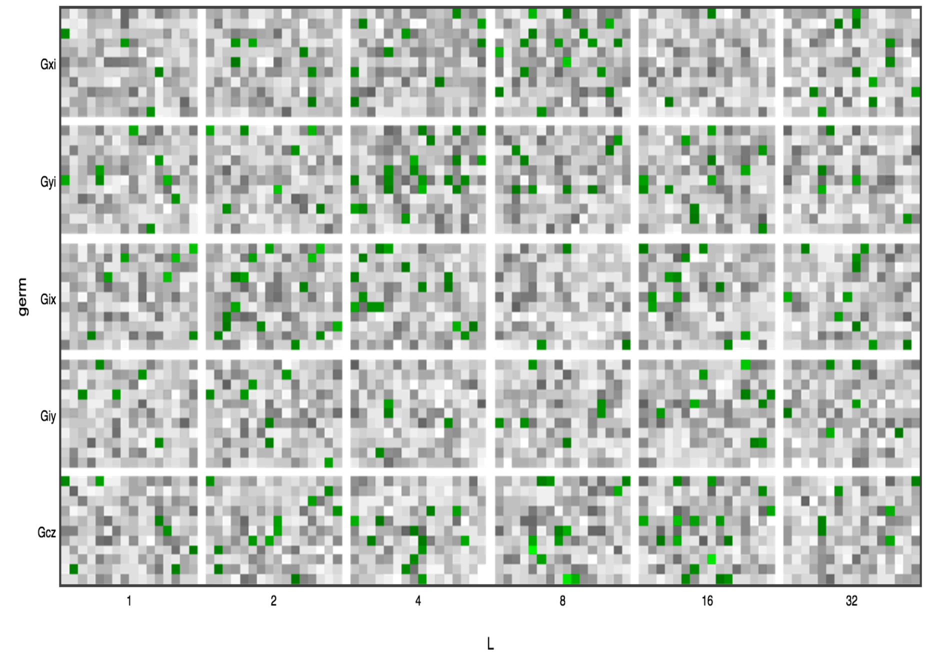
# Drift detection in Rigetti device

Simulated Markovian GST data




Simulated data where the bias *does **not** change* across sweeps (8010 gate strings)

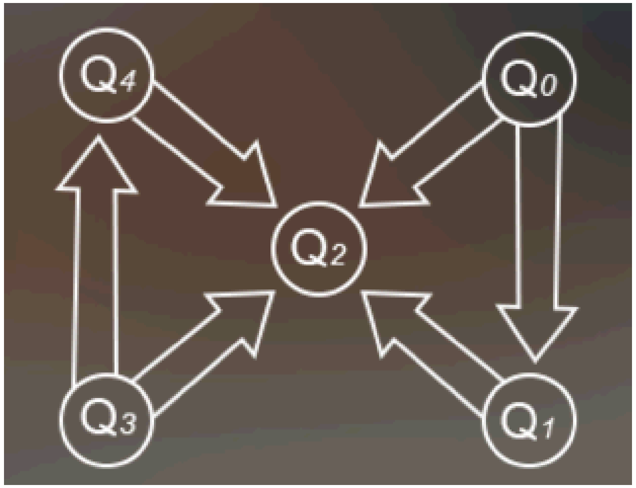
Experimental GST data (Rigetti)




Experimental data where bias ***does** change* across sweeps (8010 gate strings).
"1 in ∞" chance the gate sets are the same.

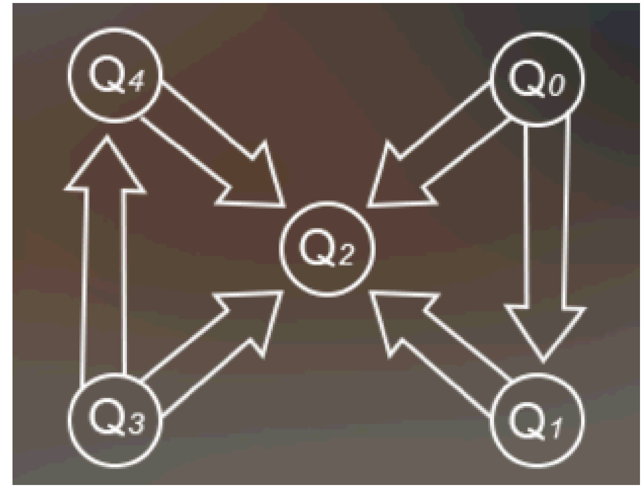
# Operation crosstalk on IBM QX2

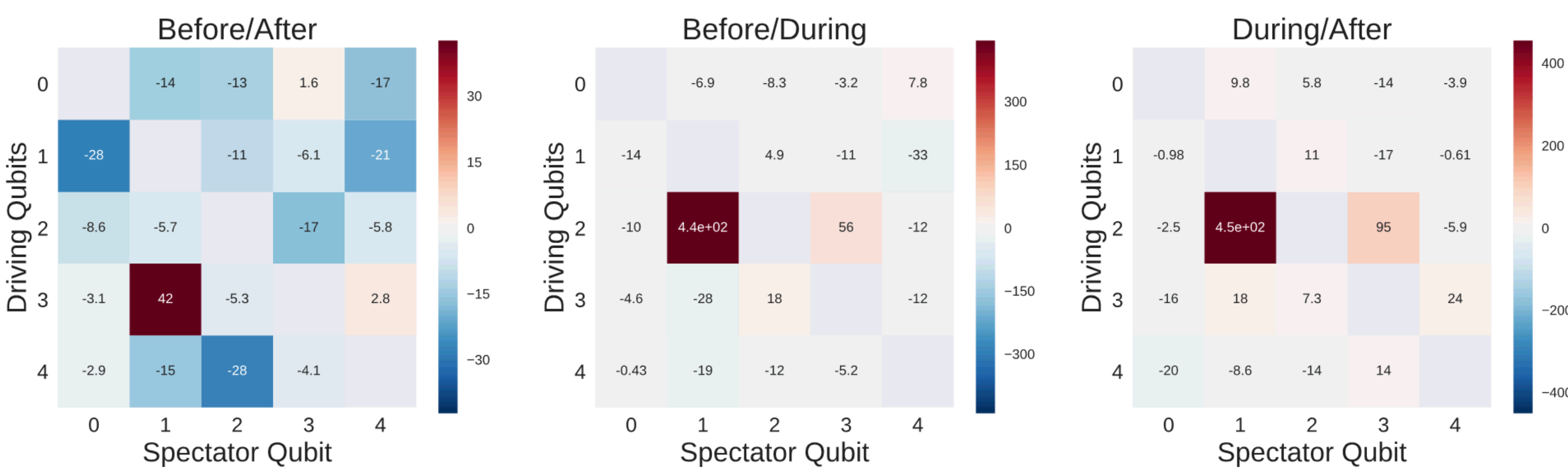
- Experiment A (Before)
  - LGST sequences on qubit A
  - *Idle gates on qubit* B
- Experiment B (During)
  - LGST sequences on qubit A
  - *Driving qubit* B *with* H *gates*
- Experiment C (After)
  - LGST sequences on qubit A
  - *Idle gates on qubit* B



# Operation crosstalk on IBM QX2

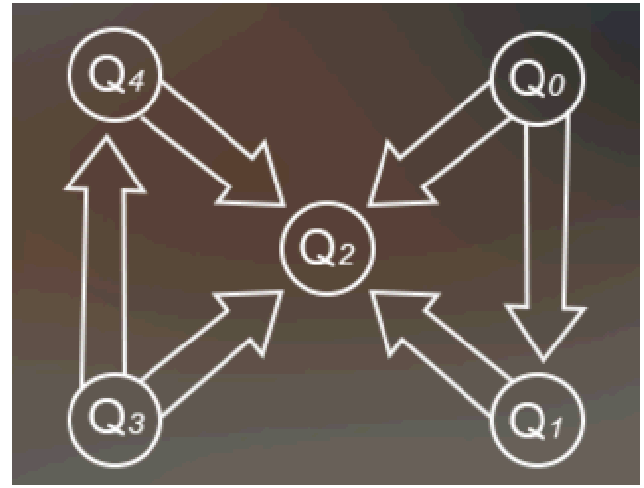
- Experiment A (Before)
  - LGST sequences on qubit A
  - *Idle gates on qubit* B
- Experiment B (During)
  - LGST sequences on qubit A
  - *Driving qubit* B *with* H *gates*
- Experiment C (After)
  - LGST sequences on qubit A
  - *Idle gates on qubit* B

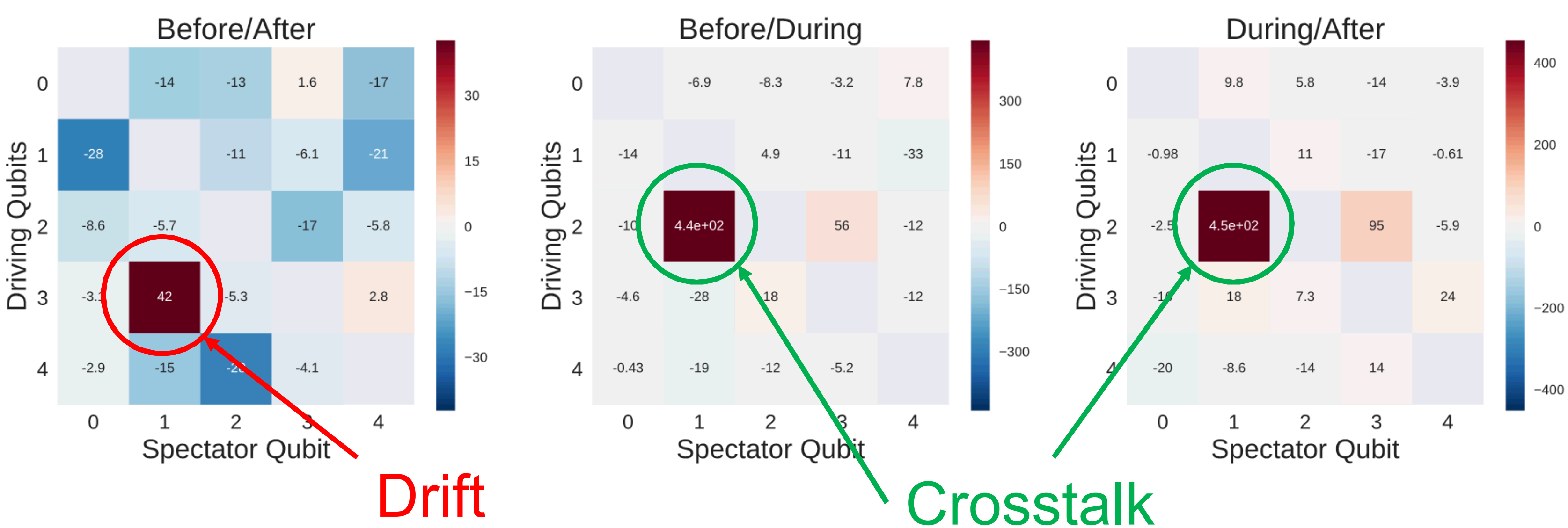


# Operation crosstalk on IBM QX2

- Experiment A (Before)
  - LGST sequences on qubit A
  - *Idle gates on qubit* B
- Experiment B (During)
  - LGST sequences on qubit A
  - *Driving qubit* B *with* H *gates*
- Experiment C (After)
  - LGST sequences on qubit A
  - *Idle gates on qubit* B



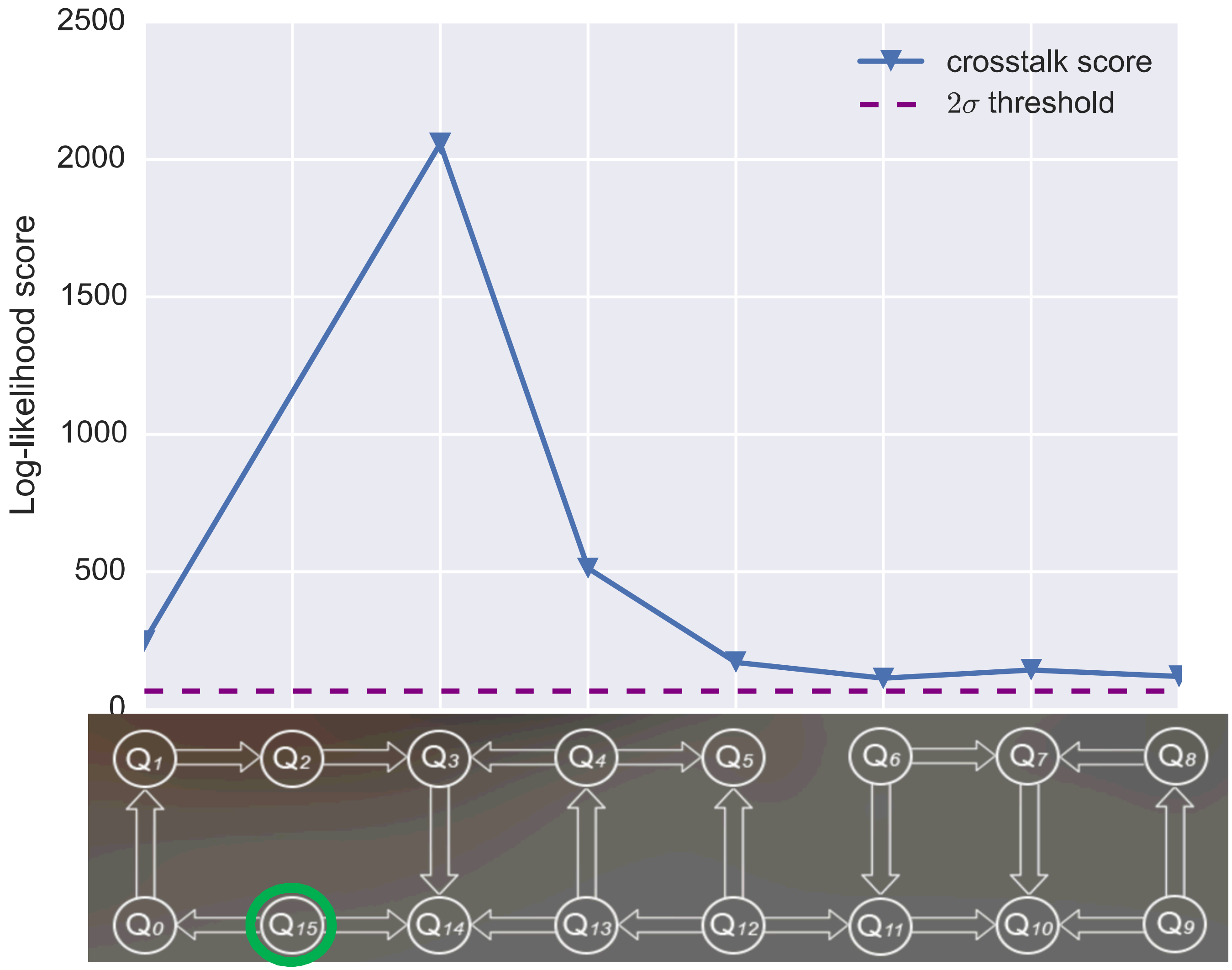



# Operation crosstalk on IBM QX2

- Drive CNOTs on ladder rungs
- Measure $Q_{15}$

Large violations of null hypothesis with minimal drift.

*Violations due to crosstalk.*



# These tools are part of pyGSTi

- You will need the beta branch of the software

```
…/pyGSTi$ git checkout beta
```

- Contained in:

```
pygsti.objects.DataComparator([ds_0,…,ds_n])
```

- Demonstrated in:

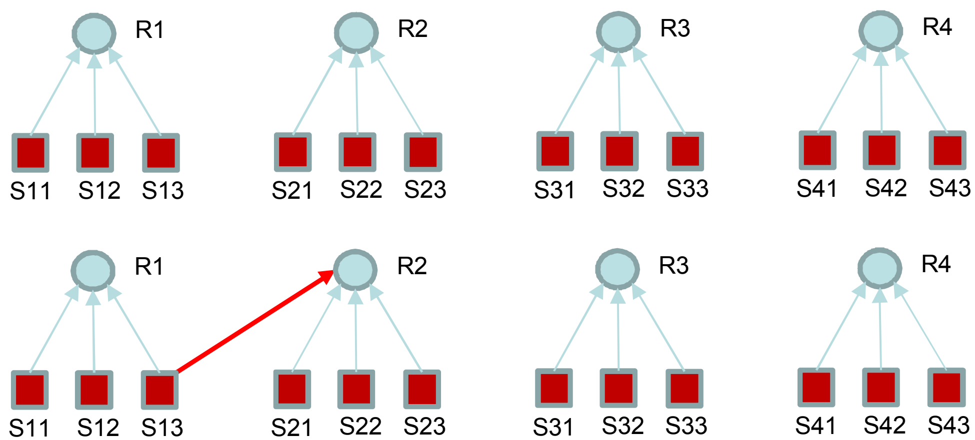
  Tutorial 17: Pure Data Analysis

- Direct questions to
  - Erik Nielsen: enielse@sandia.gov
  - Kenny Rudinger: kmrudin@sandia.gov

# Crosstalk detection via conditional independence testing

**Ideal:**

**Problematic:**



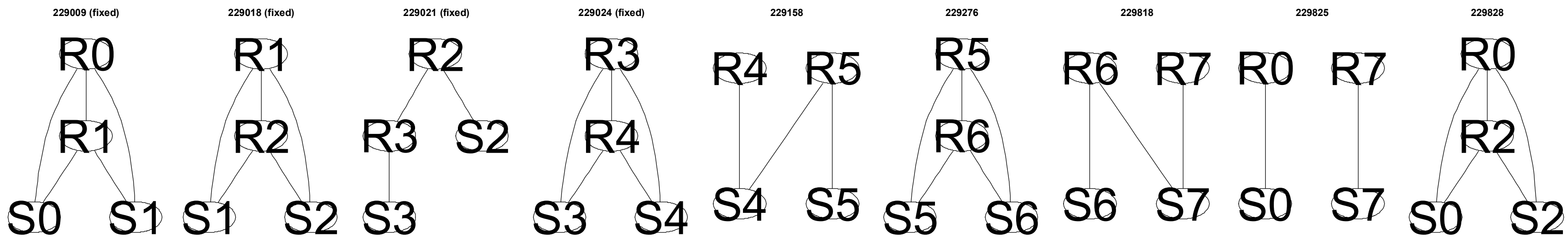

Measurement results on individual qubits

Experimental settings

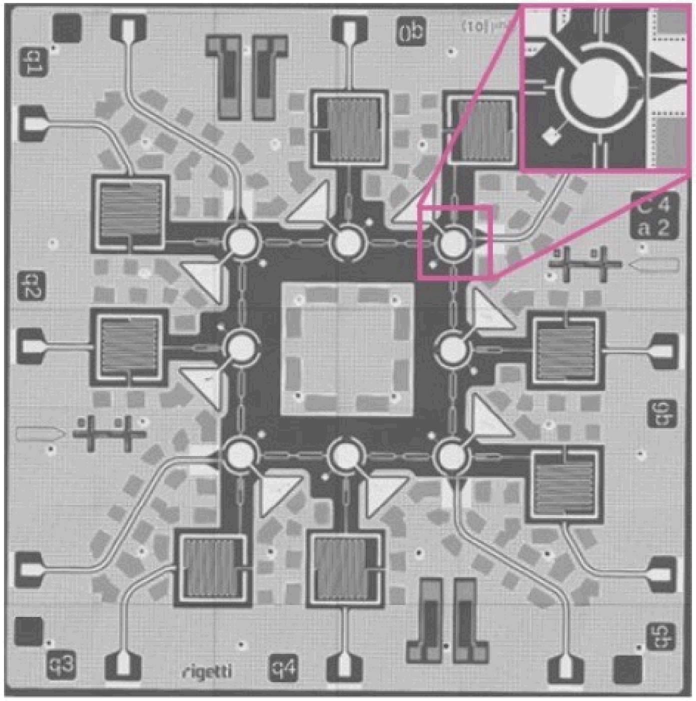
$$\mathbf{Pr}(\ R1 \mid S11, S12, S13, S21, S22, S23, \ldots) = \mathbf{Pr}(\ R1 \mid S11, S12, S13\ )$$

- A statistically motivated way to detect crosstalk. The conditional independence tests can be done at some significance level.
- The method can capture many different kinds of ”crosstalk” (and more generally, parallel context dependence) at once.
- Can use existing GST data, and is computationally efficient.

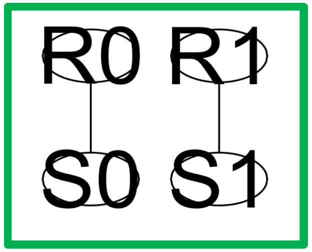
# Crosstalk detection on Rigetti device

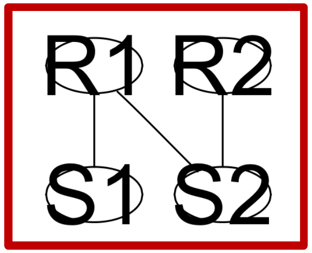


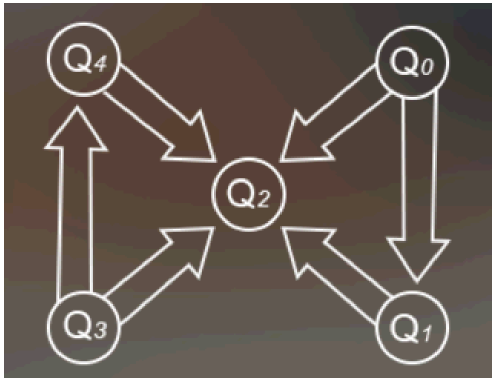

- Crosstalk detection done for all adjacent qubit pairs.
- LGST gate sequences performed on two qubits, with only the Identity germ to probe for idle crosstalk.
- *See significant crosstalk on all but* $Q_0/Q_7$



# Crosstalk detection on IBM QX2




- Experiment:
  - LGST on qubit A
  - Driving qubit B with H gates
- Crosstalk detection done for all qubit pairs.
  - Only Q1 / Q2 crosstalk detected
- IBM QX website[1] reports little coherent crosstalk between qubits 1 and 2 (ZZ interaction)
- *<u>We hypothesize</u>*: We're seeing measurement crosstalk since the resonators for these two qubits are close in frequency.

[1] https://github.com/QISKit/ibmqx-backend-information/blob/master/backends/ibmqx2/README.md

# Characterizing time dependence

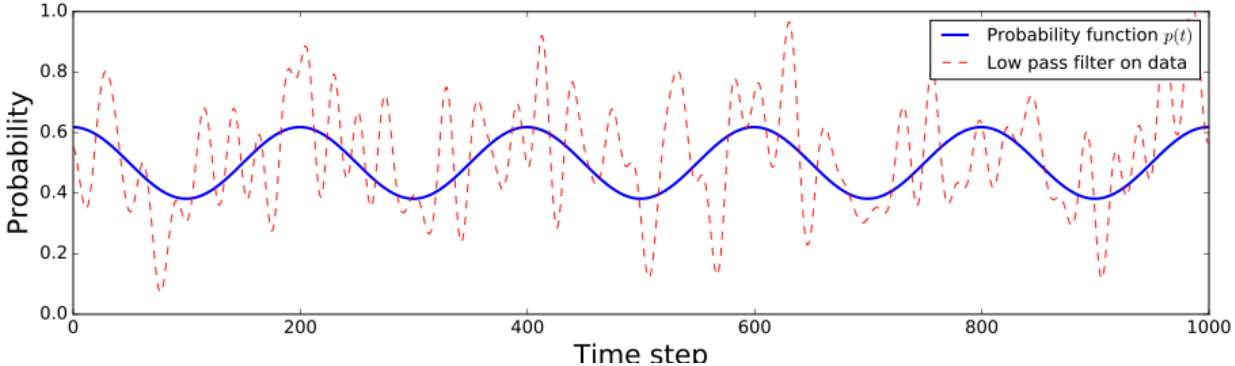
# Spectral analysis of time-dependent data

- Data is a length *N* bit string representing equally-spaced measurements:

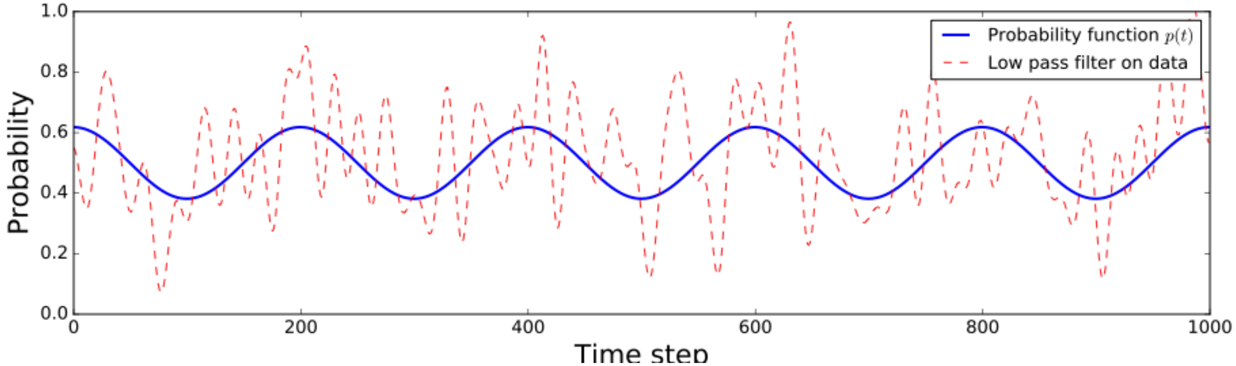
  $$\mathbf{x} = \{\, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, \ldots \,\}$$

- Hypothesis testing:
  - H0: Data drawn from *N* independent flips of a coin with a time-independent (but unknown) bias *p*.
  - H1: The bias p(t) is a non-constant function of time.
- *N* bits of information. So can only hope to detect *p(t)* described by $k \ll N$ parameters.
- We consider the most physically relevant case: **Fourier-sparse** *p(t)*.

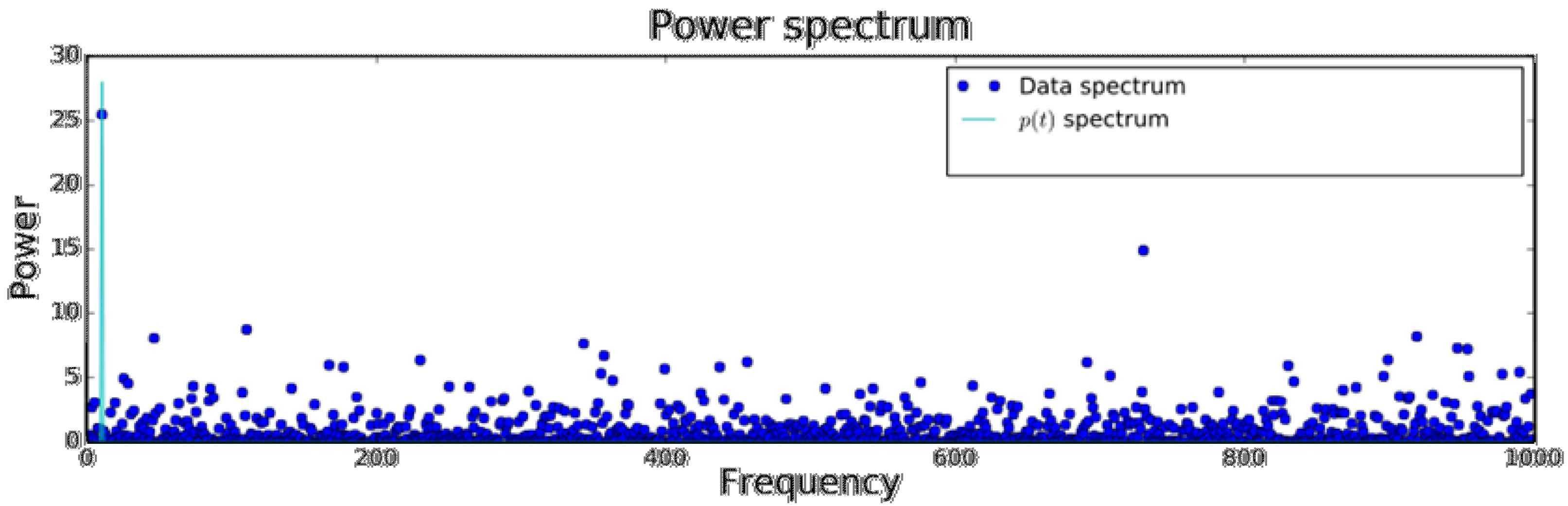
# Spectral analysis of time-dependent data – *monochromatic drift*



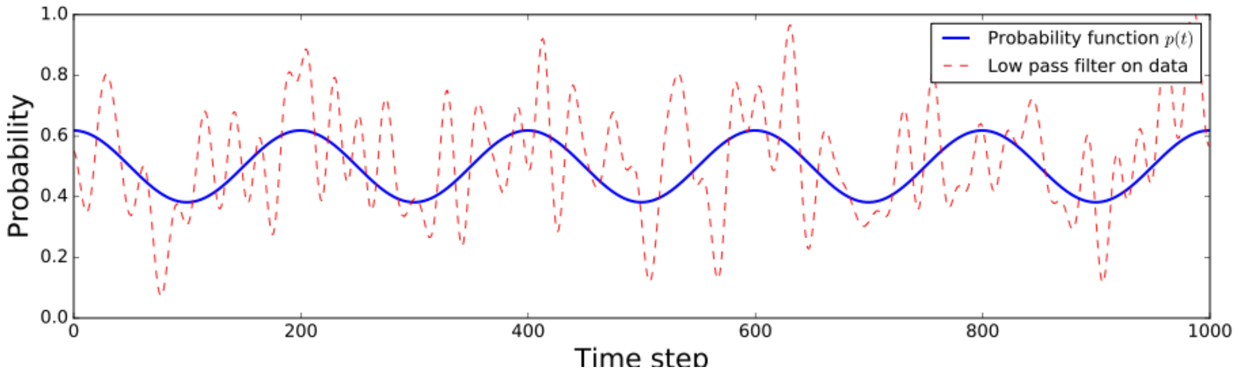
# Spectral analysis of time-dependent data – *monochromatic drift*

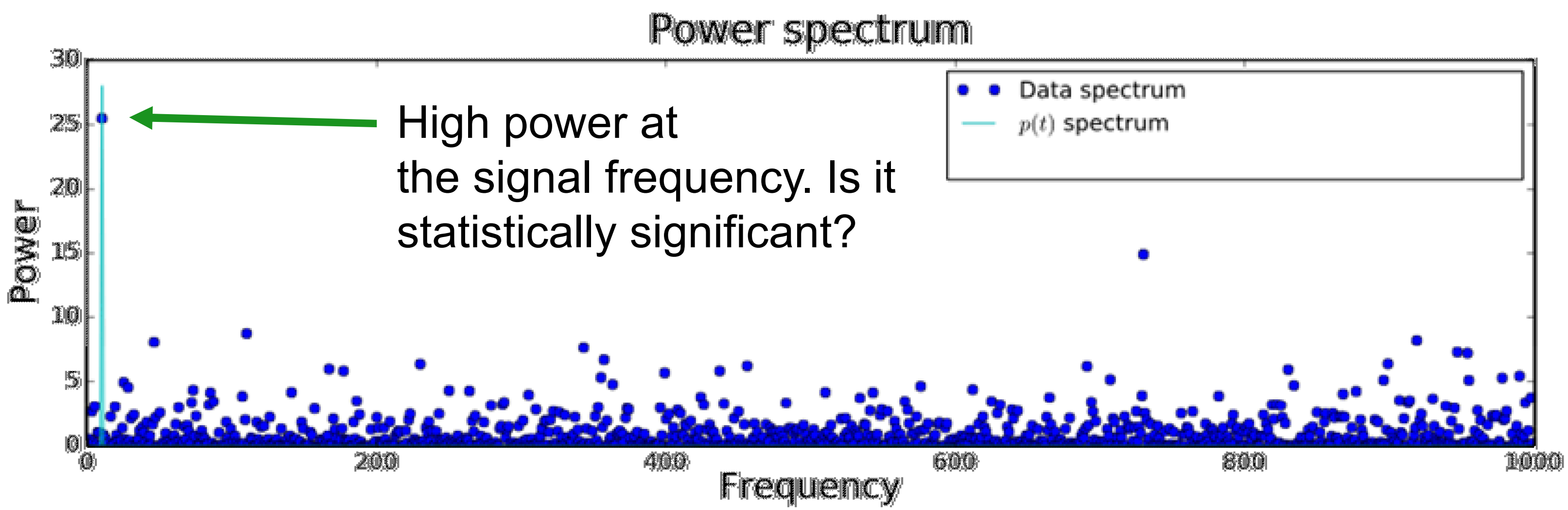



*Transform to Fourier domain*

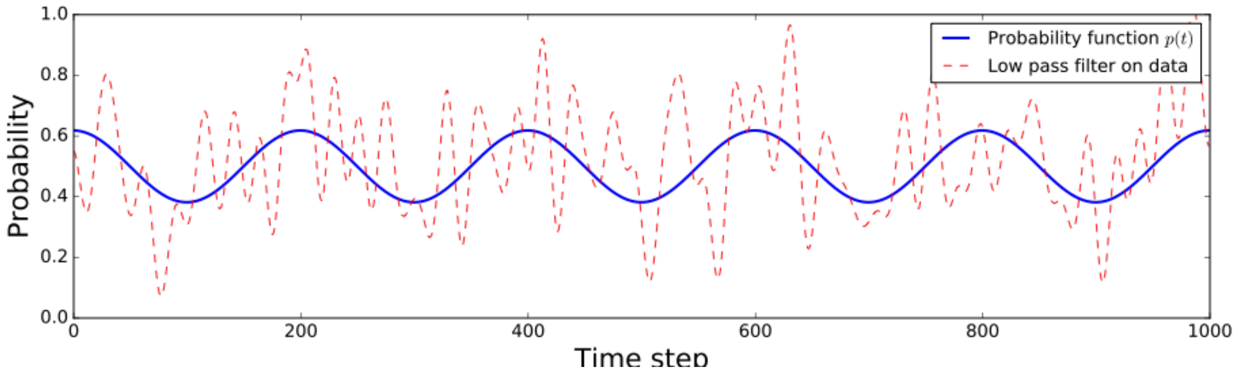


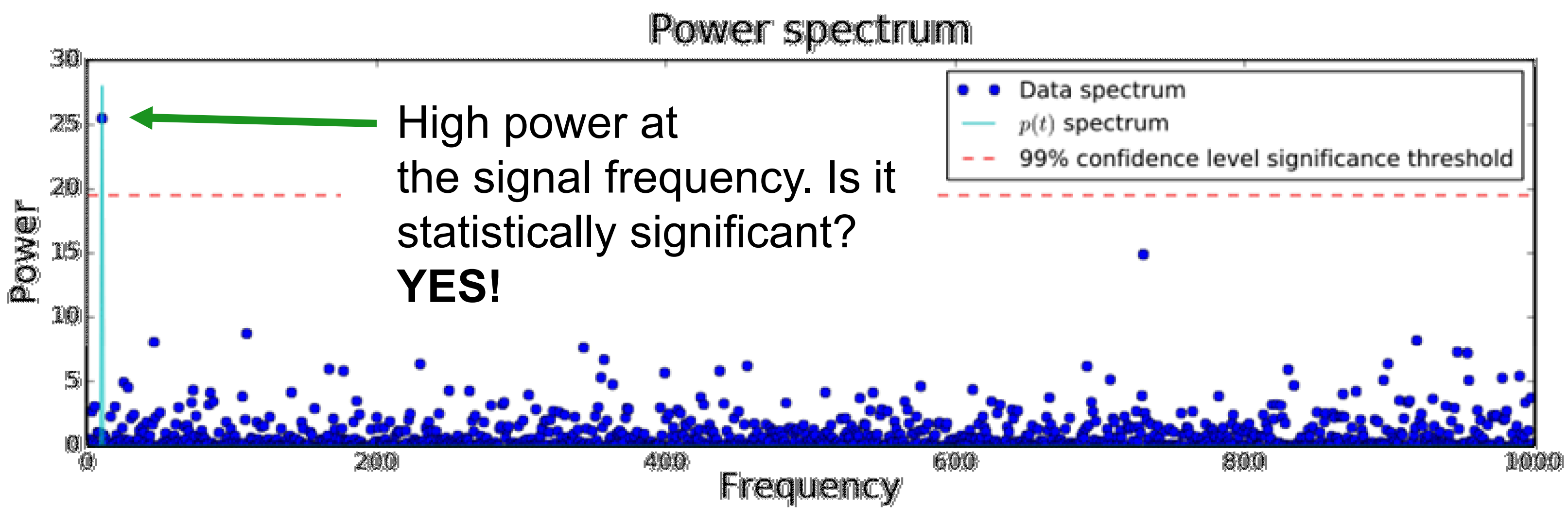
# Spectral analysis of time-dependent data – *monochromatic drift*



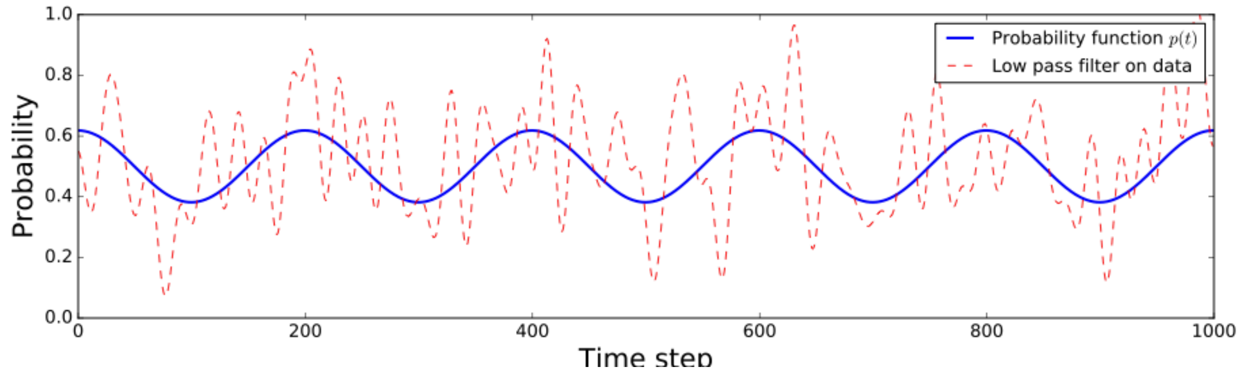



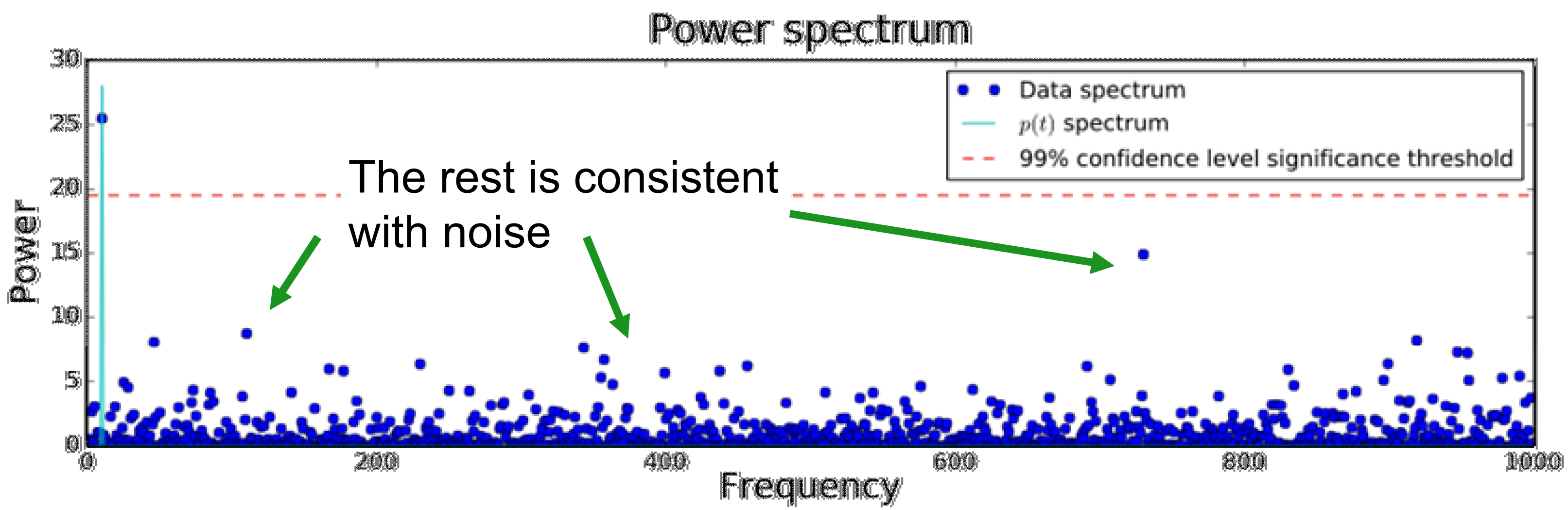
# Spectral analysis of time-dependent data – *monochromatic drift*



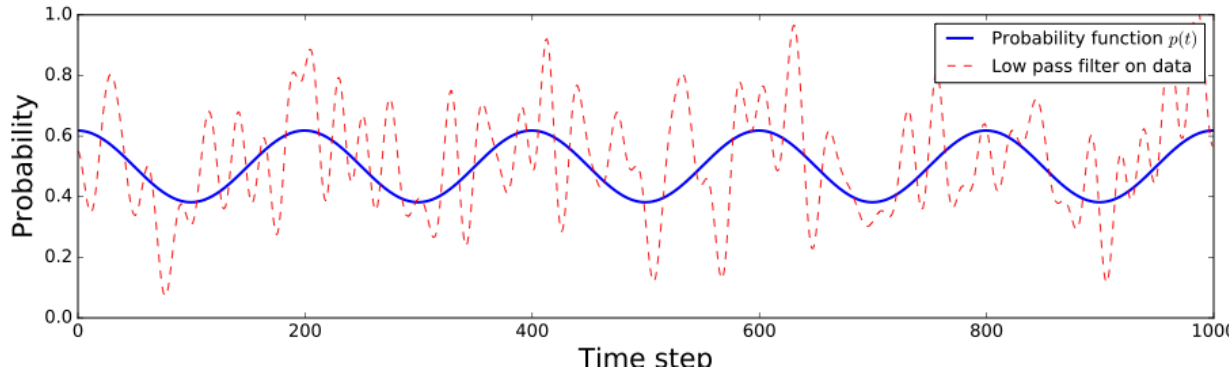



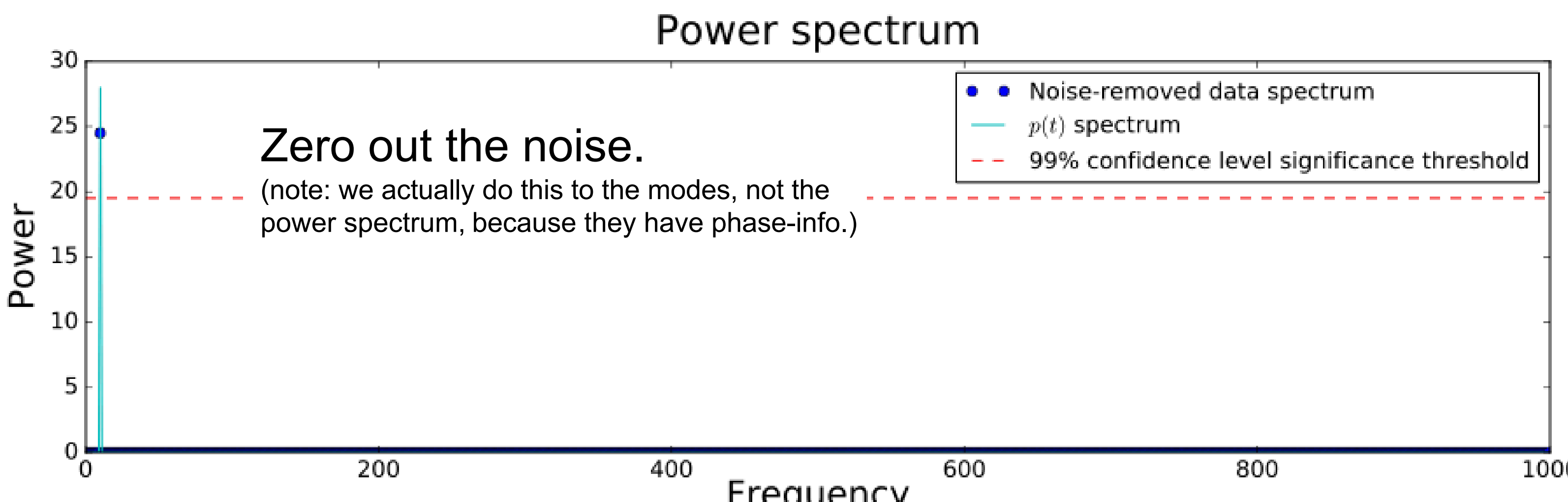
# Spectral analysis of time-dependent data – *monochromatic drift*



# Spectral analysis of time-dependent data – *monochromatic drift*

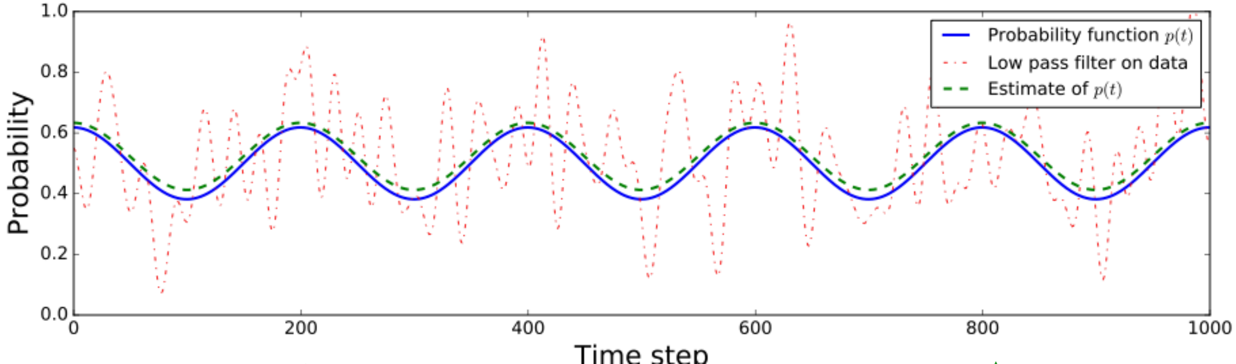


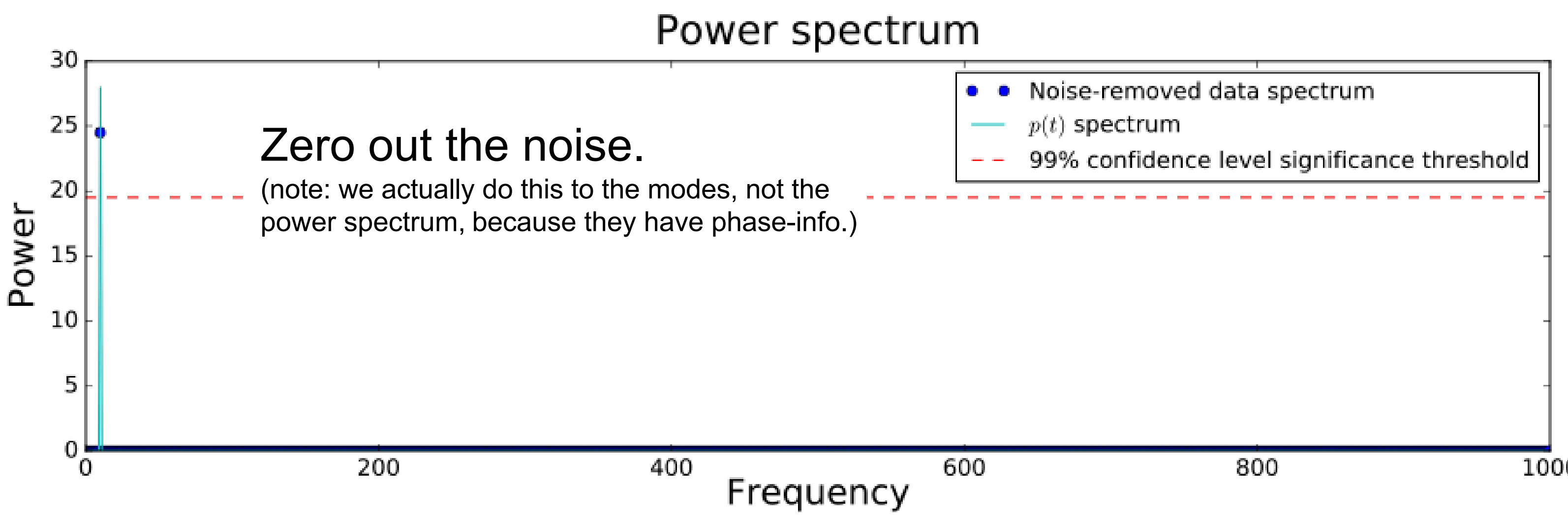

Zero out the noise.

(note: we actually do this to the modes, not the power spectrum, because they have phase-info.)

# Spectral analysis of time-dependent data – *monochromatic drift*



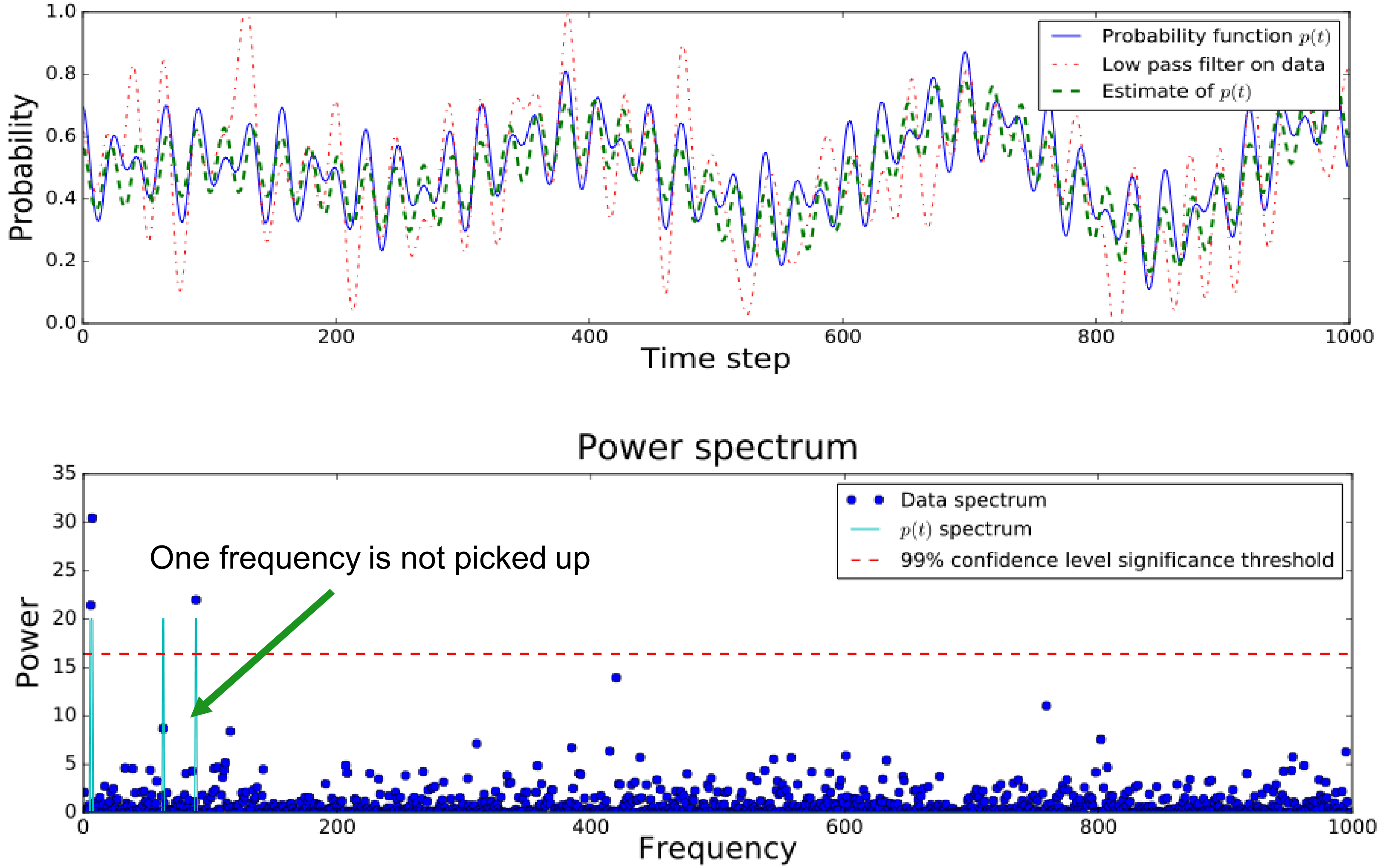

*Invert the Fourier transform*



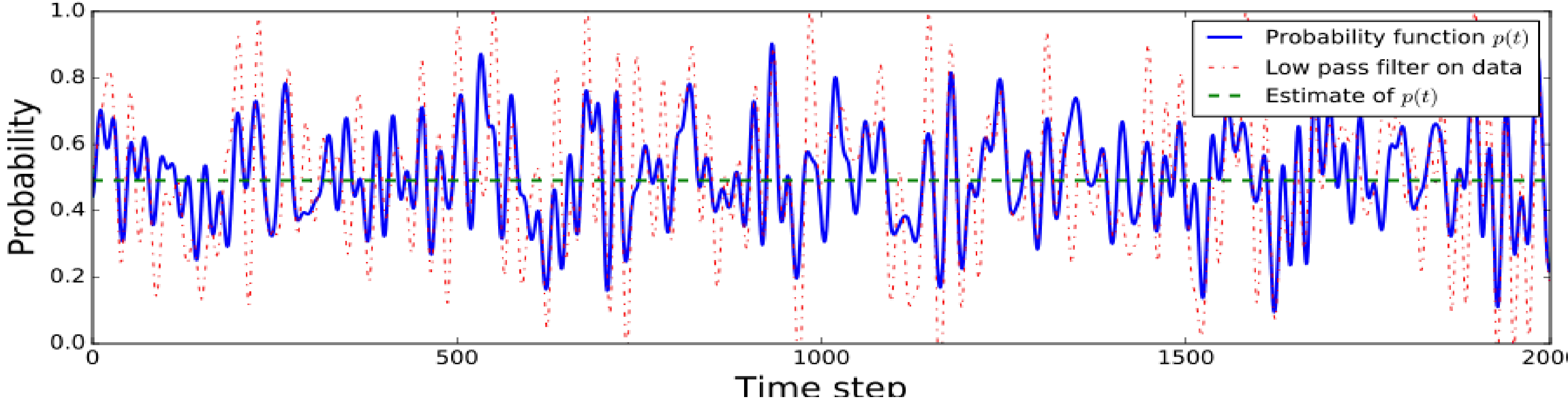

Zero out the noise.

(note: we actually do this to the modes, not the power spectrum, because they have phase-info.)

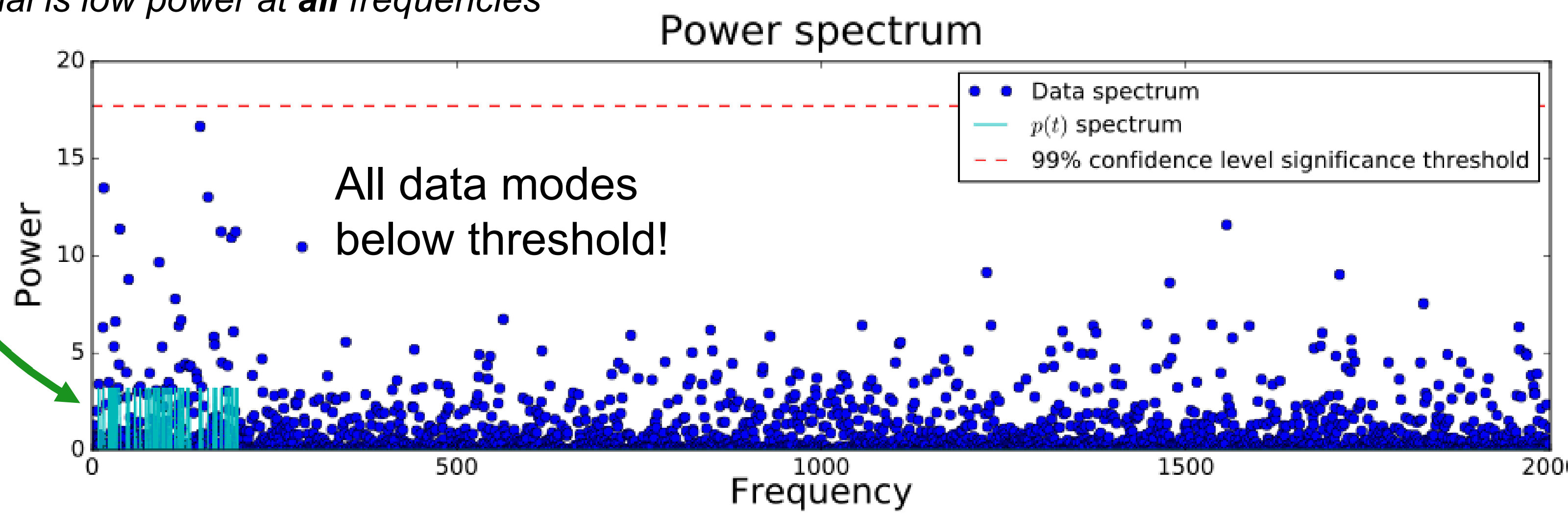
# Spectral analysis of time-dependent data – *polychromatic drift*



# Spectral analysis of time-dependent data – *general drift*

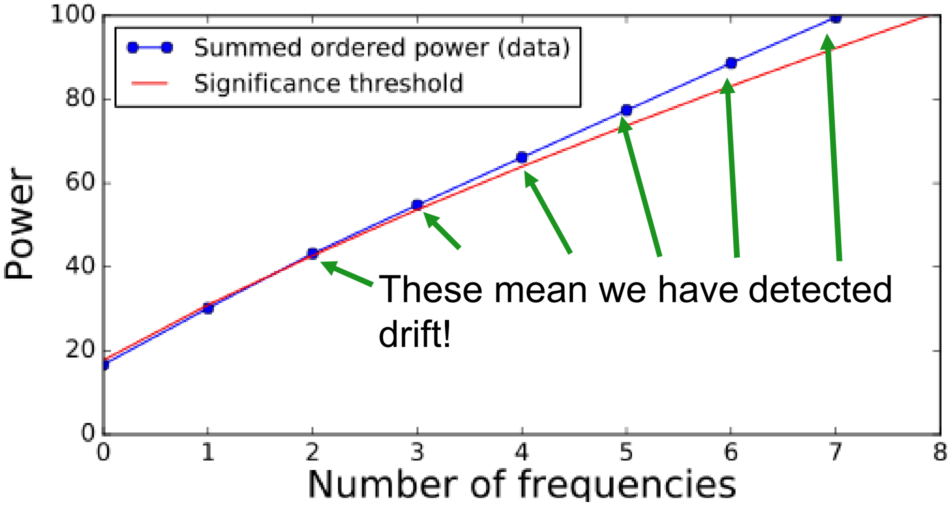



*Signal is low power at **all** frequencies*

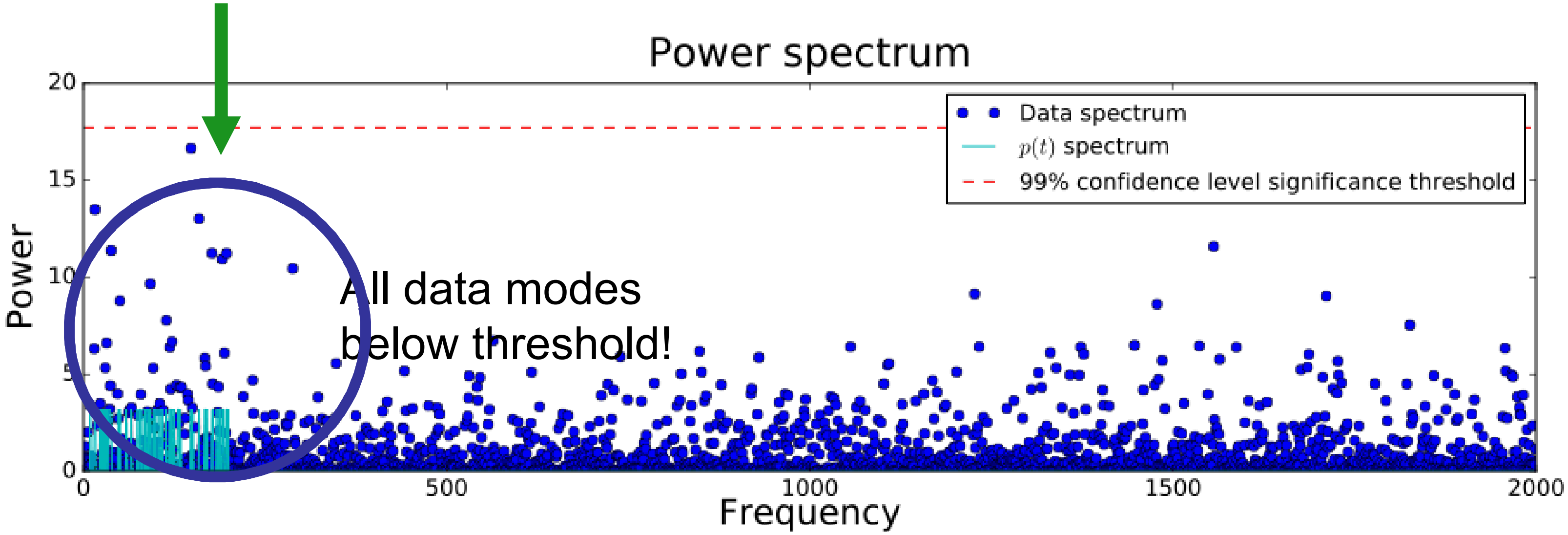


# Spectral analysis of time-dependent data – *general drift*

Summed power test statistic

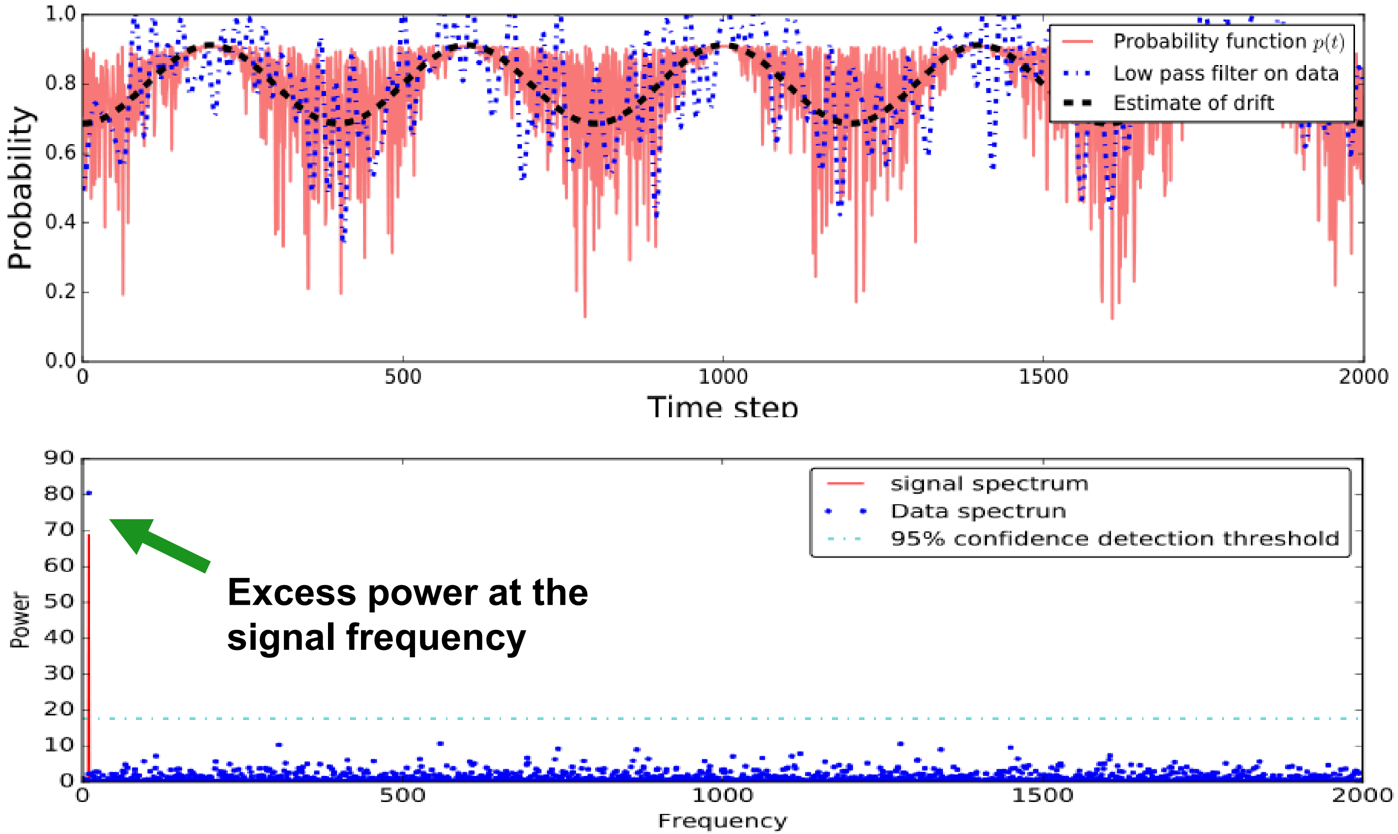



There is a lot more high-ish power modes than we would expect with no drift.



# Spectral analysis of time-dependent data – *randomized benchmarking*

- Error model: each Clifford followed by unitary $U(t) = \exp(-i\cos(\omega t)\sigma_z)$
- Each sequence performed exactly once.



# Spectral analysis of time-dependent data – *GST*

- This analysis assumes GST sequences are *rastered*

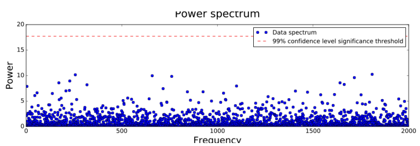
Time →

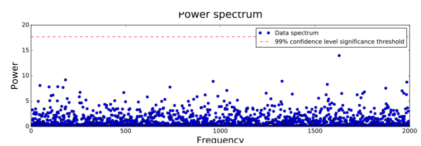
| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Circuit 1 outcomes* | 0 | | | | … | 0 | | | | … | 1 | | | |
| *Circuit 2 outcomes* | | 1 | | | … | | 0 | | | … | | 1 | | |
| *Circuit 3 outcomes* | | | 0 | | … | | | 0 | | … | | | 1 | |
| *Circuit 4 outcomes* | | | | 0 | … | | | | 1 | … | | | | 1 |

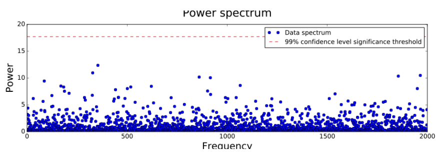
- Noise model: Hamiltonian noise in gates fluctuating at several low frequencies

# Spectral analysis of time-dependent data – *GST*

- The power spectrum for any given sequence may not be sufficient to detect drift

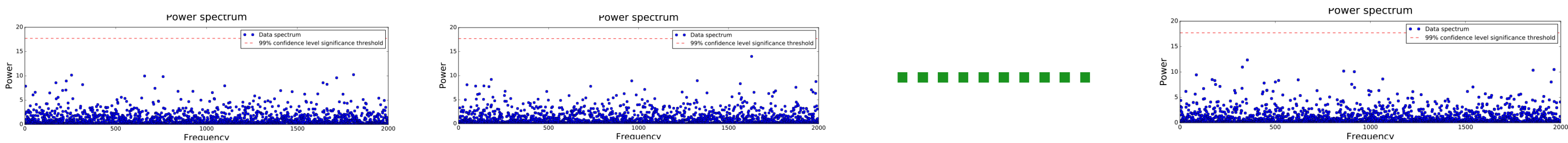


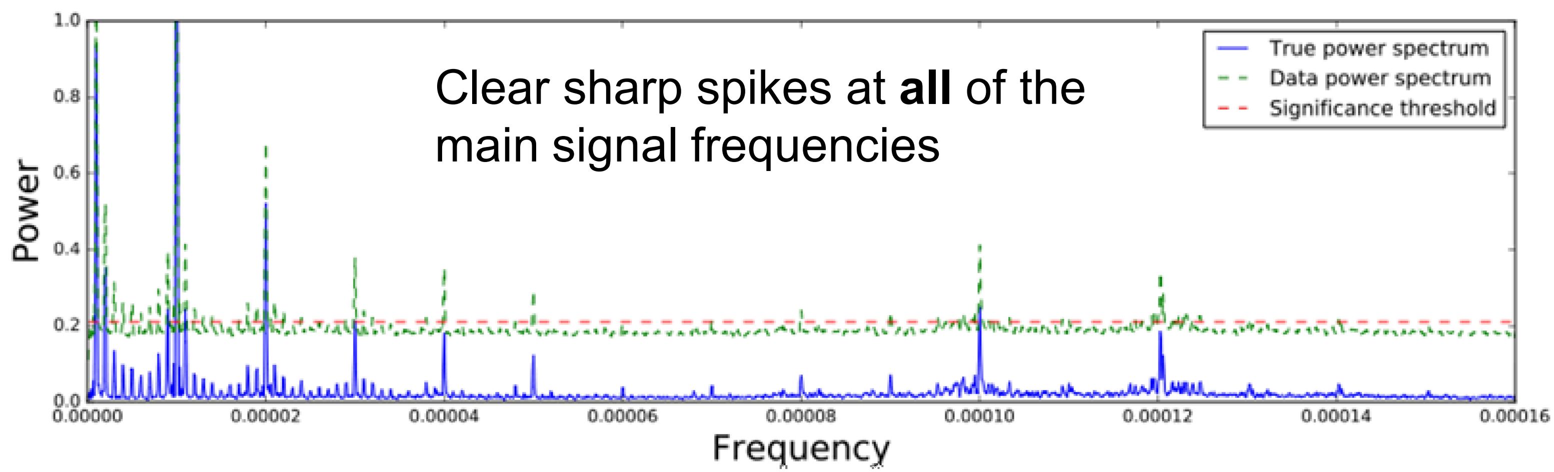



# Spectral analysis of time-dependent data – *GST*

- The power spectrum for any given sequence may not be sufficient to detect drift




- But we can average all ~2000 power spectra
  - Amplifies the signal and suppresses the noise
  - Many frequencies appear due to harmonics and aliasing




Clear sharp spikes at **all** of the main signal frequencies

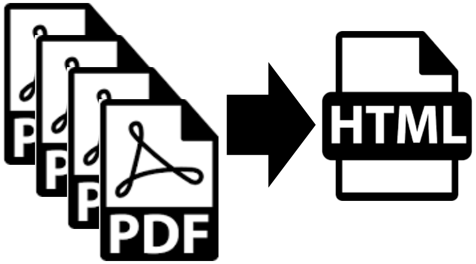
# Recent updates to pyGSTi

# Recent (beta-level) pyGSTi updates

- Using these under-development features requires that you be on the **'beta' branch of pyGSTi**

## Better data visualization (HTML reports)

- **Problem**: pyGSTi's reports required lots of scrolling and flipping between different PDF files to compare analyses.
- **Solution**: now reports are HTML documents!
- Allows **interactive** reports & denser data
- Effectively "layers" of PDF reports folded into one
  - Multiple gauge optimizations
  - Multiple models
  - Multiple data sets



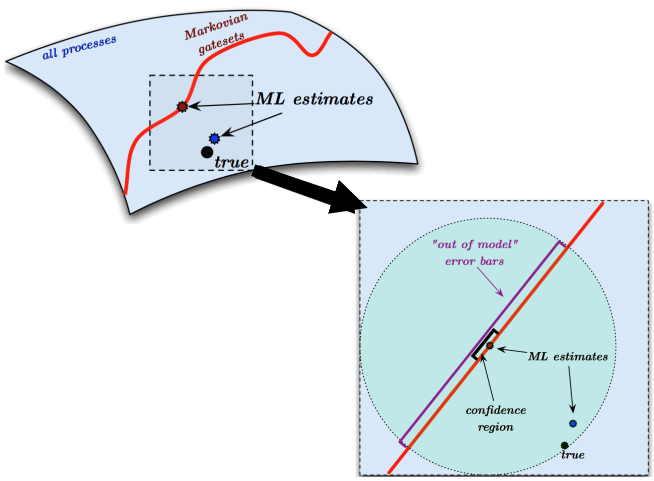
# Recent pyGSTi updates (cont)

## Non-Markovian error bars

- **Problem**: how to convey uncertainty in a gate set when the usual GST estimate cannot fit the data well (i.e. there's "non-markovian" errors) .

- **Solution (in progress)**: define some type of "non-markovian error bars". Currently we have two ways of doing this:
  - Type1: error bars based on distance from truth
  - Type2: normal in-model error bars based on data with *artificially reduced counts* so that it can be fit as well as expected.


black = sequence omitted,
white = sequence kept



# Recent pyGSTi updates (cont)

**Standard practice GST** – convenient interface for multiple GST estimates

- **Problem1**: sometimes GST will give gate estimates that are not CPTP, and this invalidates metrics such as the process fidelity (can get process infidelities < 0, etc).

- **Problem2**: knowing the "right" parameters for gauge optimization is difficult, and GST analyses can be difficult to interpret because it's difficult to separate gauge artifacts from actual gate errors.

- **Solution**: We've tried to bottle several "standard" ways of running GST which includes:
  - both CPTP and TP fits
  - Multiple gauge optimizations
  - Automatic computation of non-Markovian error bars when needed.

- To run GST using these standard-practice setting, just call `do_stdpractice_gst` instead of `do_long_sequence_gst`.

- Integrates nicely with HTML reports, which are able to display multiple estimates and gauge optimizations at once.

# Acknowledgements

- Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525