Ben Barsdell
Matthew Bailes
Christopher Fluke
David Barnes

Spotting Radio Transients

With the help of GPUs

Background

- High Time Resolution Universe (HTRU) survey
  - Running since 2008, now entering deep phase
- Uses the Parkes 64m radio telescope
  - Located in remote NSW, Australia

- Goal is to discover new pulsars and radio transients
  - (And diamond planets!)

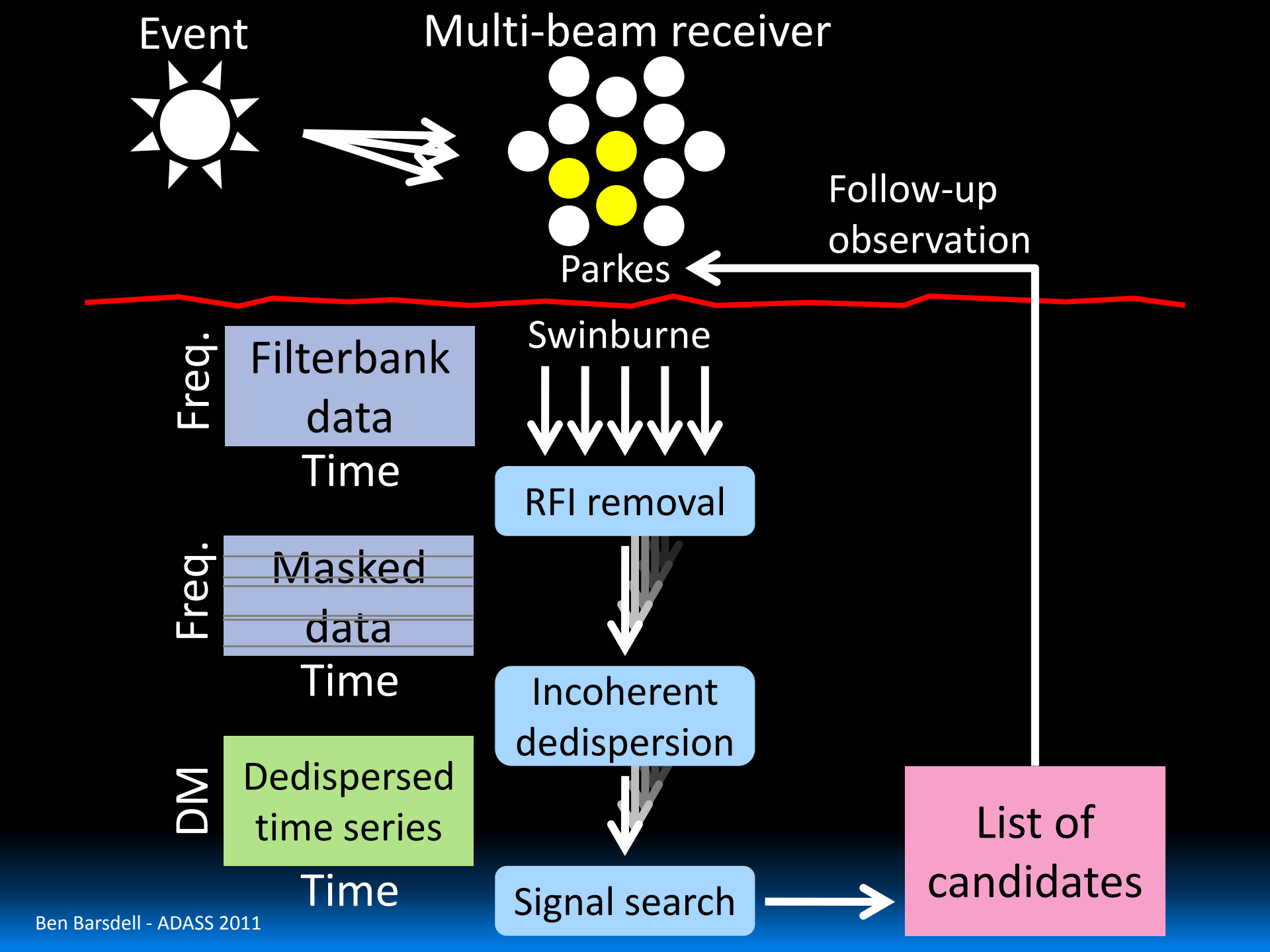
Survey specs

400 MHz BW @ 1381.8 MHz

1024 freq. channels

64µs time resolution

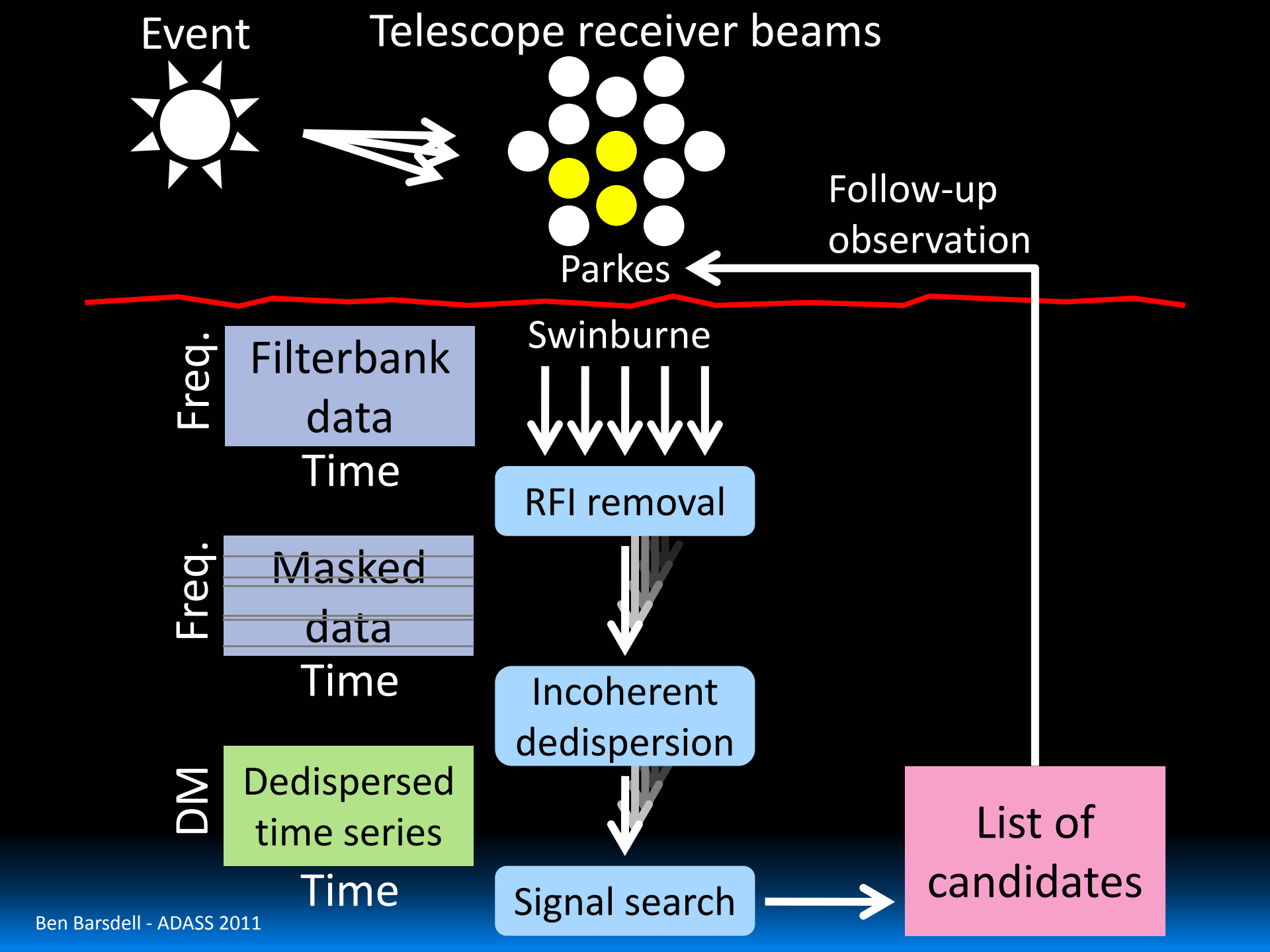
2-bit sampling


Event
Multi-beam receiver
Follow-up observation
Parkes
Swinburne
Freq.
Filterbank data
Time
RFI removal
Freq.
Masked data
Time
Incoherent dedispersion
DM
Dedispersed time series
Time
Signal search
List of candidates

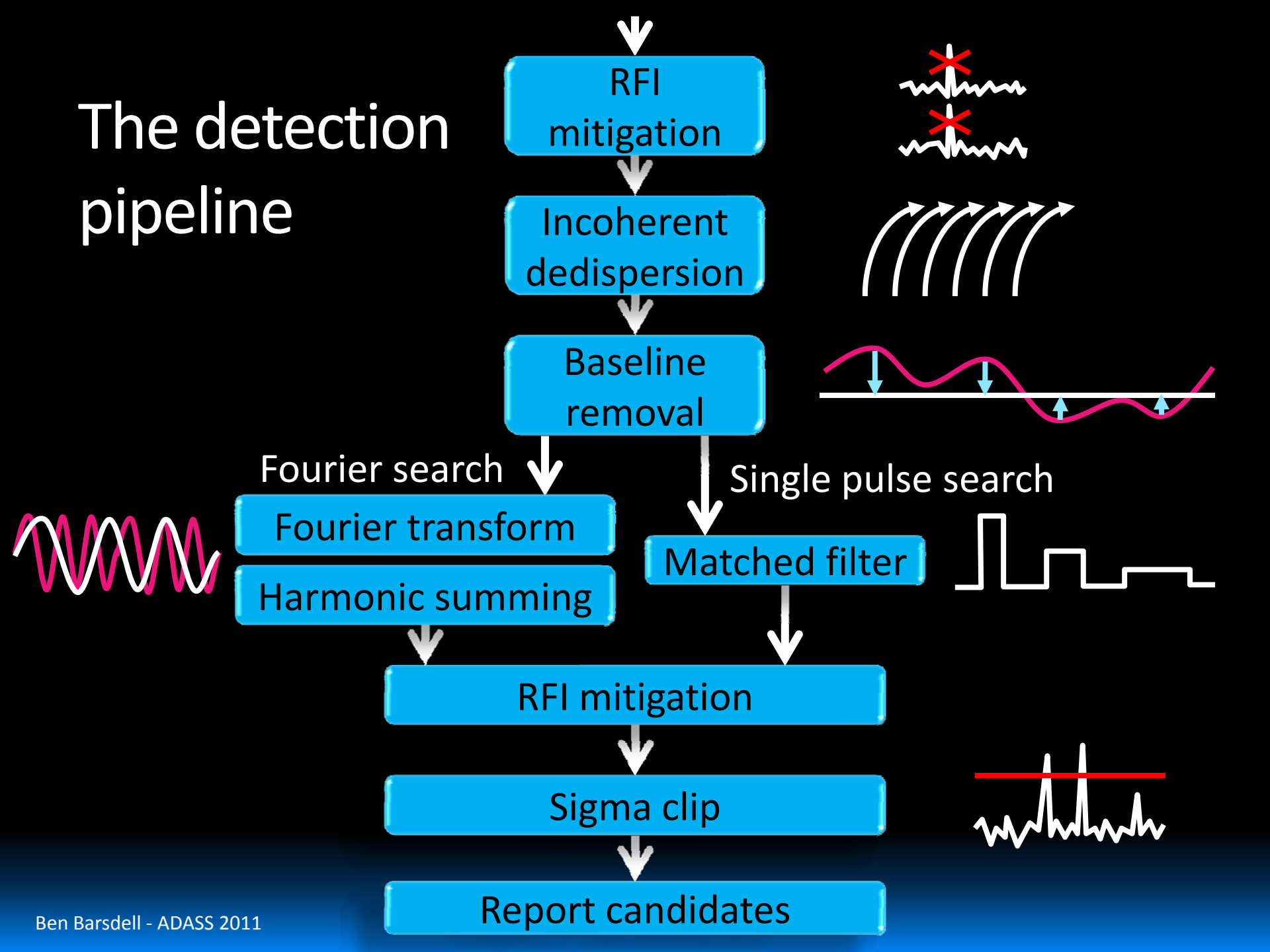
The Plan

- Current pipeline takes
  > 30 mins per 10 min observation
  - Necessitates off-line processing
  - Means transfers, tapes and long waits
- Would like to speed things up to **real-time**
  - Instant feedback and follow-up observations
  - Triggered baseband data dumps
- How to do it?
- Time to bring out the heavy artillery…

The GPU


Event
Telescope receiver beams
Follow-up observation
Parkes
Swinburne
Freq.
Filterbank data
Time
RFI removal
Freq.
Masked data
Time
Incoherent dedispersion
DM
Dedispersed time series
Time
Signal search
List of candidates

The detection pipeline



RFI mitigation

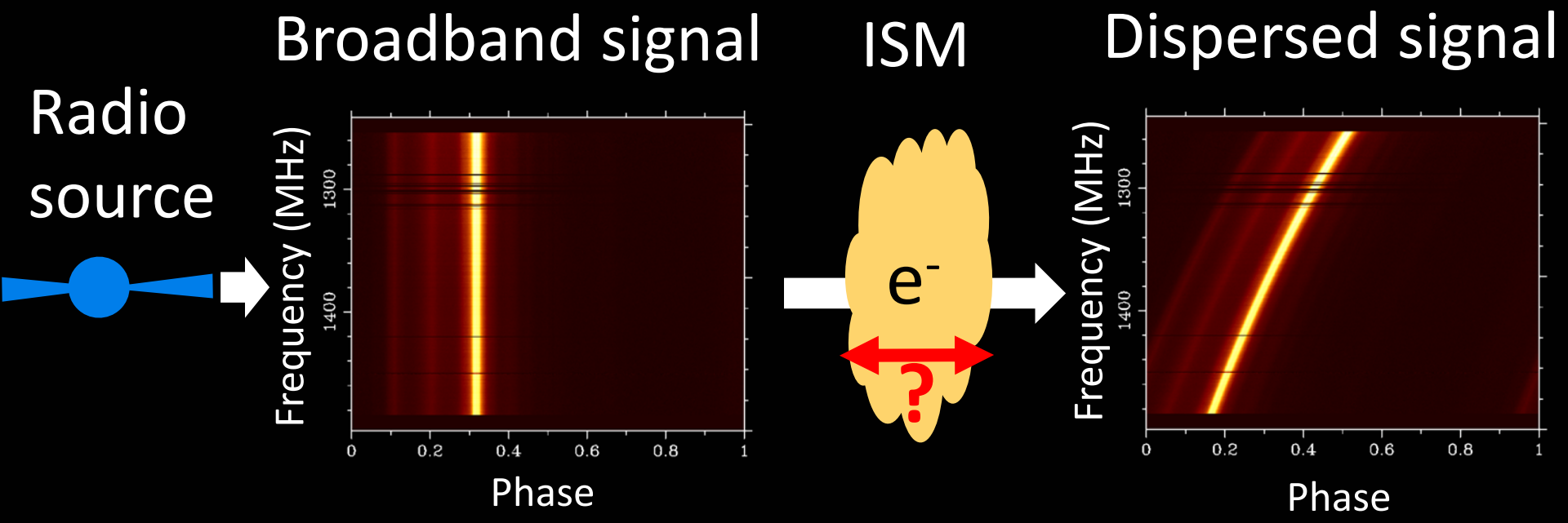
- Interference is a big problem
- No easy solution
  - Military radar too useful
  - Prime-time TV too popular
- Some things can be done
  - Sigma clipping
  - Spectral kurtosis
  - Coincidence rejection

www.clker.com

Coincidence rejection

- Use multi-beam receiver as reference antennas
  - Assume RFI is not localised
- Apply simple coincidence criteria:
  - E.g., 3σ in 4+ beams => RFI
- Or use Eigen-decomposition approach
- Run on GPU as a straightforward transform
  - `RFI_mask[i] = is_RFI(multibeam_data[i])`
  - Note: Eigen-decomp method makes `is_RFI()` trickier

Dedispersion



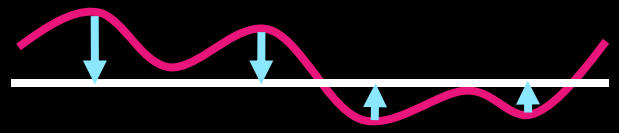
- Unknown distance => search through DM space
- Pick DM, dedisperse, search, repeat
  - ~1200 DM trials

Dedispersion

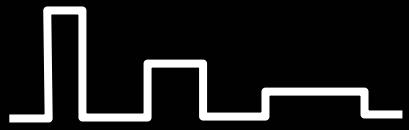
- Computationally intensive problem
  - Biggest time-consumer
- Runs really well on a GPU
  - Lots of parallelism
  - High arithmetic intensity
  - Good memory access patterns
  - No branching

Other algorithms

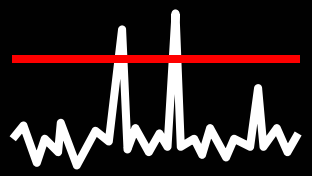
- Baseline removal
  - Subtract running mean
  - Port to GPU using *parallel prefix sum*



- Matched filtering
  - Convolve with 1D boxcar
  - Can also use parallel prefix sum



- Sigma cut + peak find
  - Threshold and segment
  - Port to GPU using segmented reduction



Preliminary results

- Dedispersion: 20 mins → 2.5 mins
  - Using ‘direct’ method on 1 Tesla C2050 GPU
  - Details in Barsdell et al. (refereed)
- Nearly completed porting other algorithms
- Goal of 10 mins well within reach!

Looking ahead

- Real-time radio transient detection promises to
  - Simplify the data processing procedure
  - Enable immediate follow-up observations
  - Allow capture of high-resolution baseband data for significant events
  - Catch things like the ‘Lorimer burst’ as they happen!

Merci!

Hardware configuration