Title: CHIME: the Canadian Hydrogen Intensity Mapping Experiment

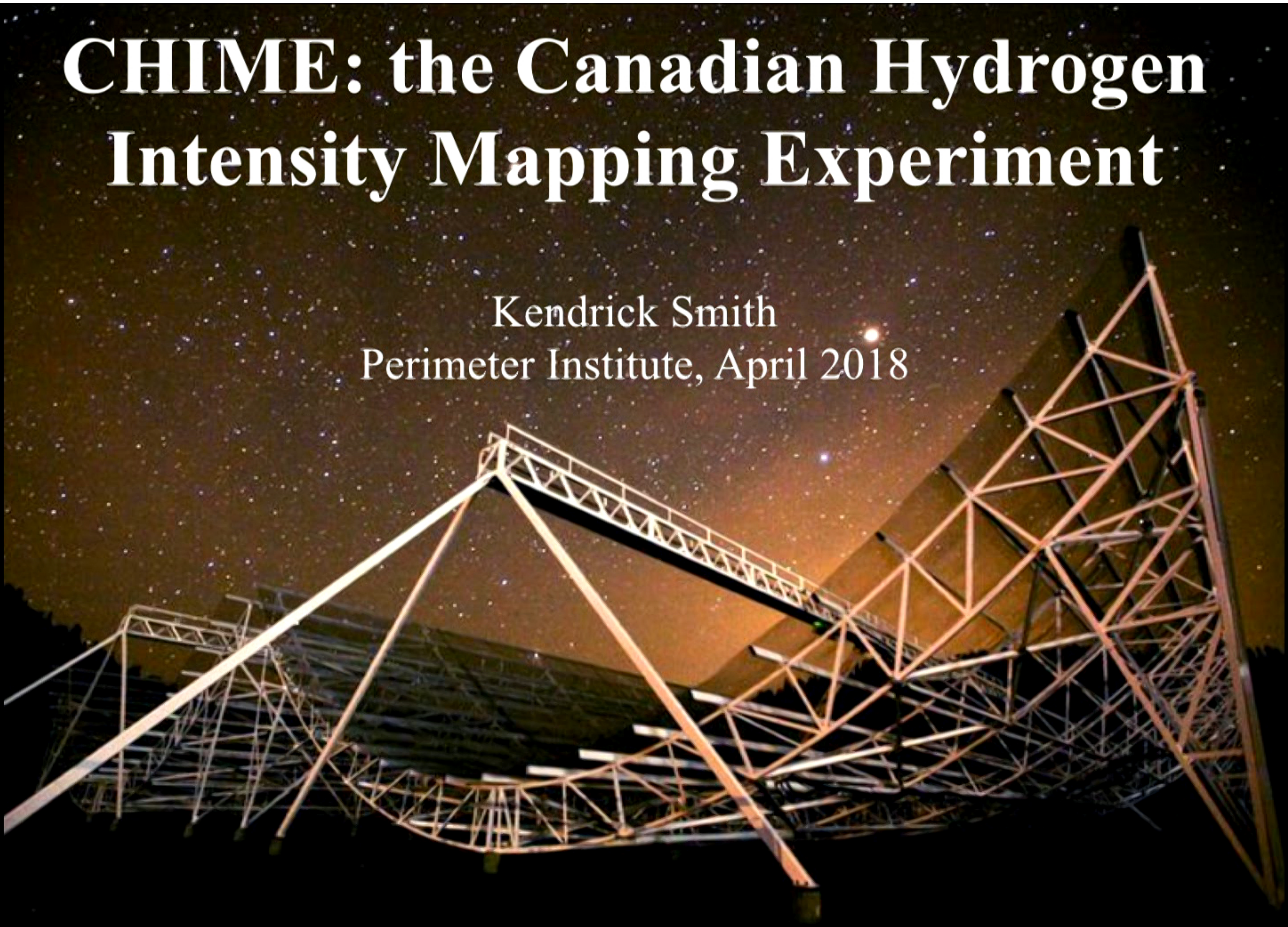
Date: Apr 04, 2018 02:00 PM

URL: http://pirsa.org/18040069

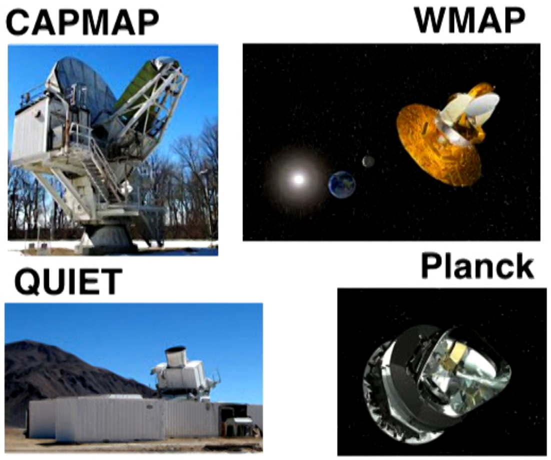
Abstract: <p>CHIME is a new interferometric telescope at radio frequencies 400-800 MHz.The mapping speed (or total statistical power) of CHIME is among the largestof any radio telescope in the world, and the technology powering CHIME could be used to build telescopes which are orders of magnitude more powerful.This breakthrough sensitivity has the power to revolutionize radio astronomy, but meeting the computational challenges will require breakthroughs on the algorithmic side. I'll give a status update on CHIME, with an emphasis on new algorithms developed at Perimeter to search for fast radio bursts (FRB's) and pulsars.</p>

CHIME: the Canadian Hydrogen Intensity Mapping Experiment

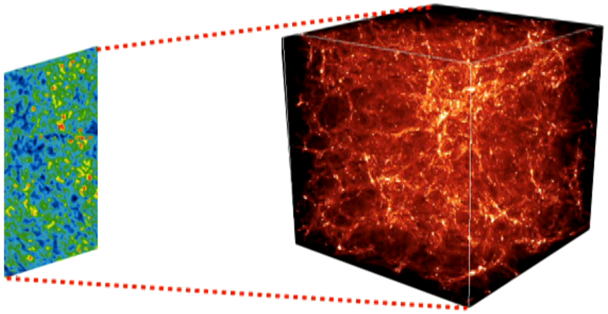
Kendrick Smith
Perimeter Institute, April 2018



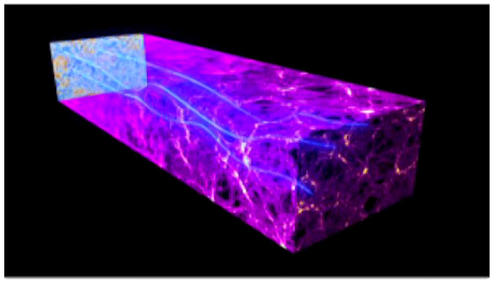
Things I'm not going to talk about



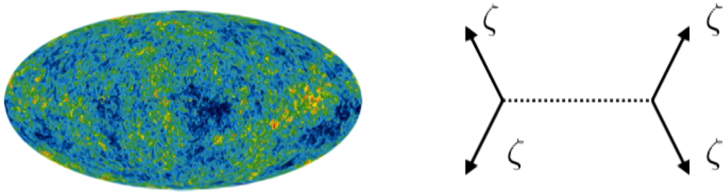
CMB experiments



Cosmology with the kinetic SZ effect



Gravitational lensing



Primordial non-Gaussianity
("cosmological collider physics")

CHIME

Lead institutions:

THE UNIVERSITY OF BRITISH COLUMBIA

McGill

UNIVERSITY OF TORONTO

DRAO

NRC · CNRC

+ Smaller teams at these institutions:

Yale University

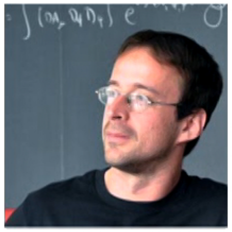
Carnegie Mellon University

West Virginia University

PERIMETER INSTITUTE

NRAO

CHIME at Perimeter

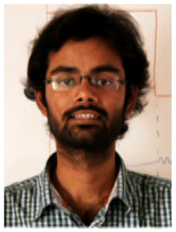


Kendrick Smith
(faculty)

Masoud Rafiei-Ravandi
(PhD student)

Maya Burhanpurkar
(undergraduate student)

Dustin Lang
(visiting postdoc)



Utkarsh Giri
(PhD student)

Canada's largest radio telescope unveiled in British Columbia

New $16M telescope an all-Canadian project between universities and National Research Council of Canada

New radio telescope unveiled in B.C.

The National

September 7, 2017 | 02:04

Scientists hope CHIME, Canada's largest radio telescope, will be a major step forward in uncovering the secrets of the universe

"Unprecedented" CHIME radio telescope completed

David Szondy | September 9th, 2017

This Huge Telescope is Built Like A Halfpipe and Hunts for Dark Energy

JACOB DUBÉ
Sep 7 2017, 11:24am

Dark energy is driving the accelerating expansion of the universe. No one knows what it is.

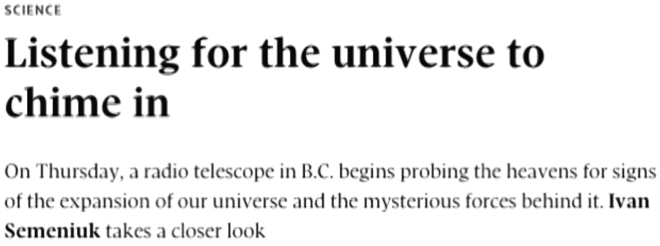
Starting bell sounds for CHIME

Sep 8, 2017

CANADIAN HYDROGEN INTENSITY MAPPING PROGRAM September 7, 2017 9:02 pm

Mapping the universe and understanding dark matter in the south Okanagan

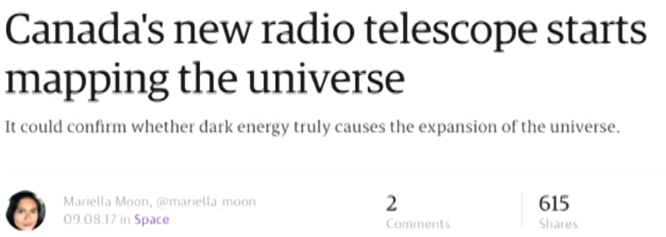
By Blaine Gaffney
Reporter Global News



SCIENCE

Listening for the universe to chime in

On Thursday, a radio telescope in B.C. begins probing the heavens for signs of the expansion of our universe and the mysterious forces behind it. **Ivan Semeniuk** takes a closer look



Canada's new radio telescope starts mapping the universe

It could confirm whether dark energy truly causes the expansion of the universe.

Mariella Moon, @mariella_moon
09.08.17 in Space

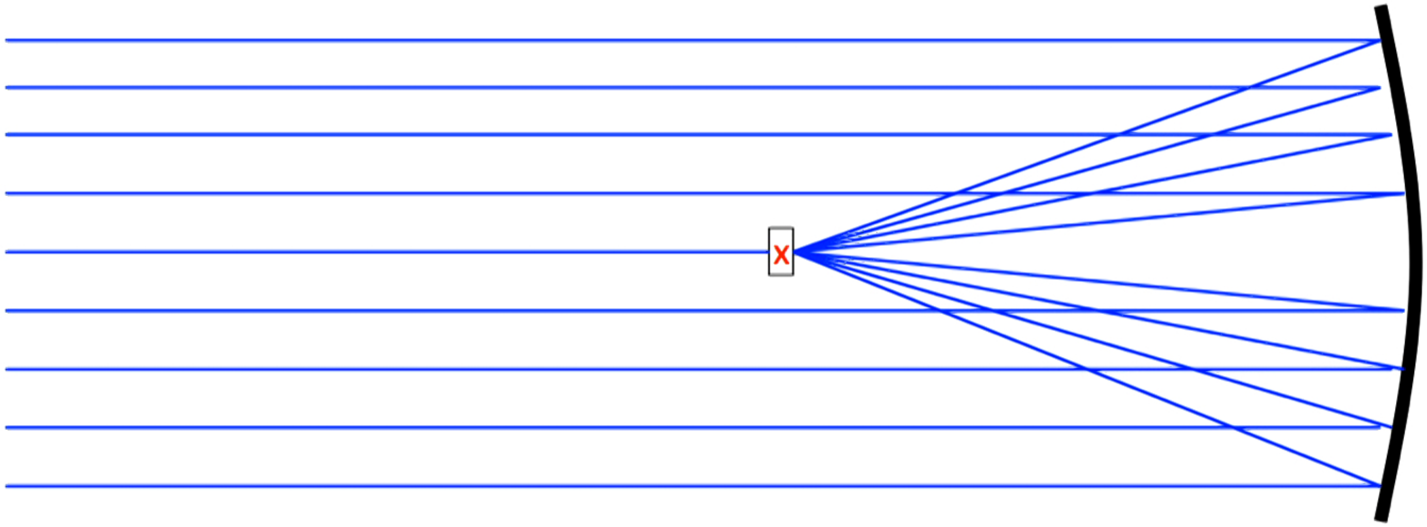
2 Comments

615 Shares

CHIME

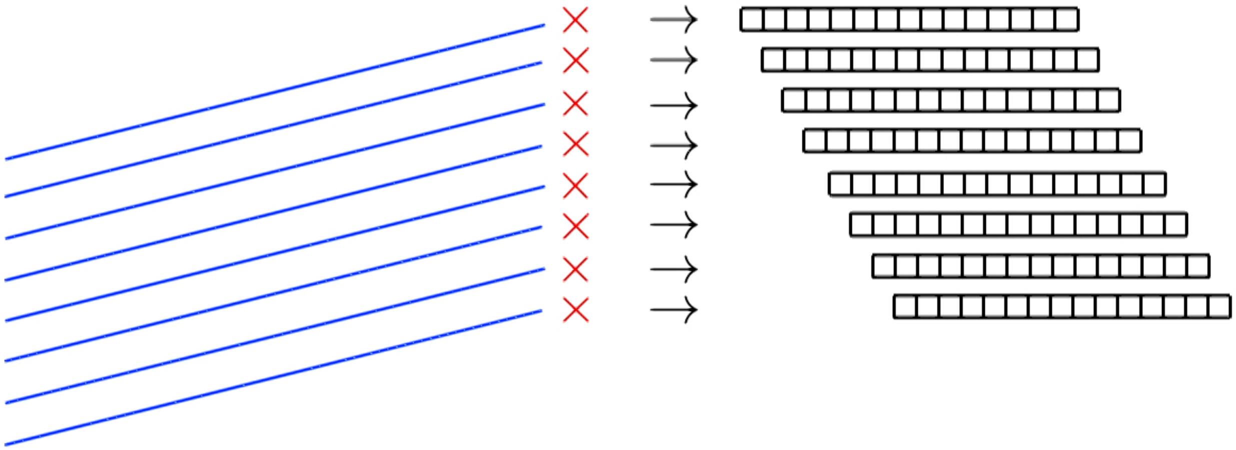
Traditional radio telescope

Single-feed radio telescope



Focuses via physical delays: constructive interference only occurs for a specific direction on the sky

Phased-array interferometer

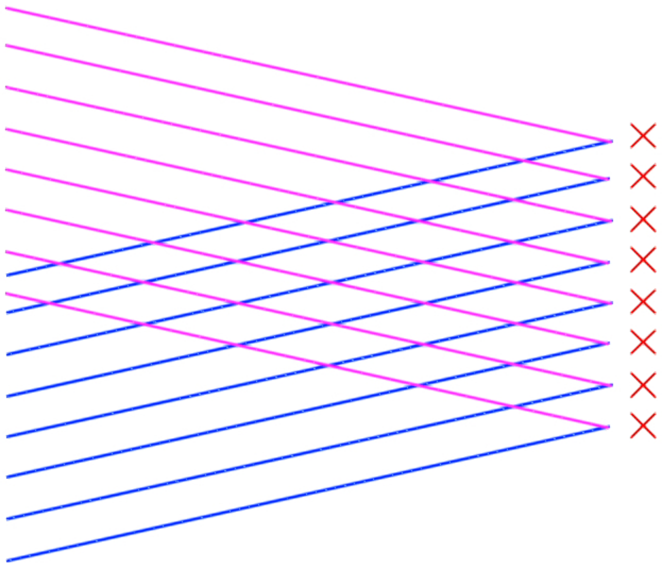


Dish is replaced by an array of antennas whose signals are digitized.

By summing signals with appropriate delays, can simulate the dish in software, and focus on part of the sky.

Can “repoint” telescope by changing delays.

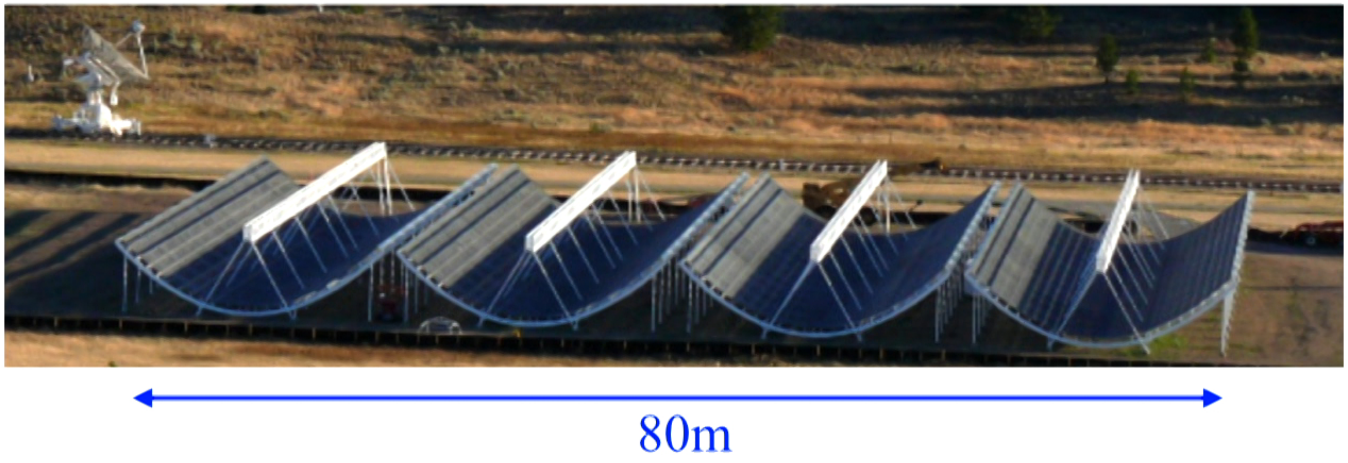
Beamforming interferometer



Copy the digitized signals and repeat the computation N times (in parallel). Equivalent to N telescopes pointed in different directions.

CHIME

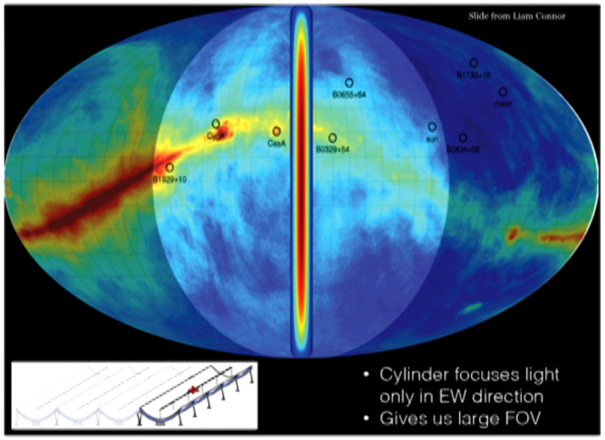
- CHIME has a 4 x 256 array of antennas and can form all 1024 independent beams in real time. Raw sensitivity is the same as 1024 single-feed radio telescopes!
- Currently in a debugging and commissioning stage.



CHIME beamforming, cartoon form

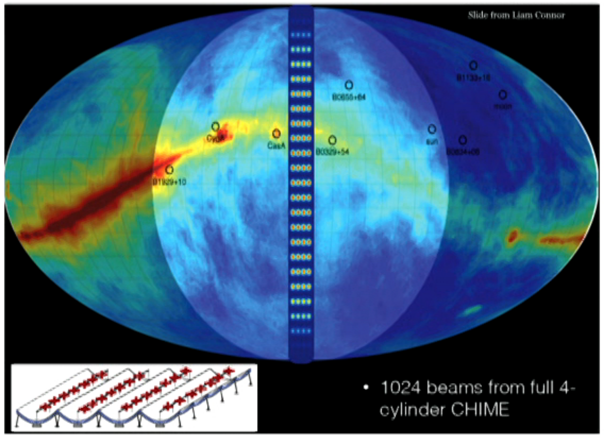
Each antenna sees a narrow strip on the sky (“primary beam”).

By beamforming in software as previously described, we can make 1024 “formed” beams with size ~0.3 degree.



- Cylinder focuses light only in EW direction
- Gives us large FOV

primary beam

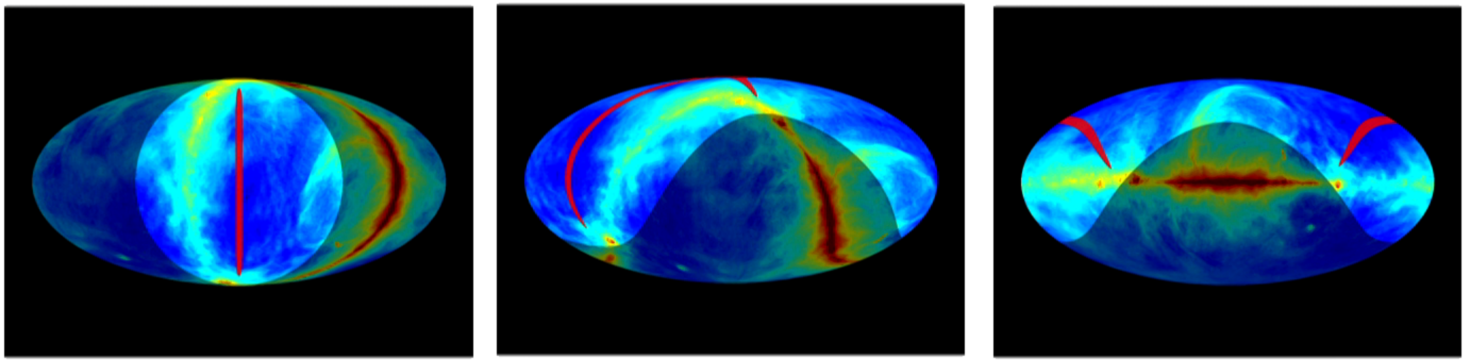


- 1024 beams from full 4-cylinder CHIME

formed beams

CHIME beamforming, cartoon form

As the Earth rotates, the primary and formed beams sweep over the sky.



Every 24 hours, we make an image of the sky with 0.3 degree resolution (= size of formed beams).

Frequency range: 400-800 MHz. Interesting for many things! (Cosmology at redshifts $0.8 < z < 2.5$, pulsars, FRB’s…)

Mapping speeds (back-of-envelope)

For many purposes, the statistical power of a radio telescope can be quantified by its mapping speed:

$$M \approx (\text{Collecting area } A) \times (\text{Number of beams}) \times (\text{order-one factors})$$

	A	N_{beams}	$M/(10^5 \text{ m}^2)$
Parkes 64m	3200 m^2	13	0.41
Green Bank 100m	7850 m^2	7	0.55
Arecibo 300m	70000 m^2	7	4.9
FAST 500m	200000 m^2	19	38
CHIME	6400 m^2	1024	66

=  CHIME ?!

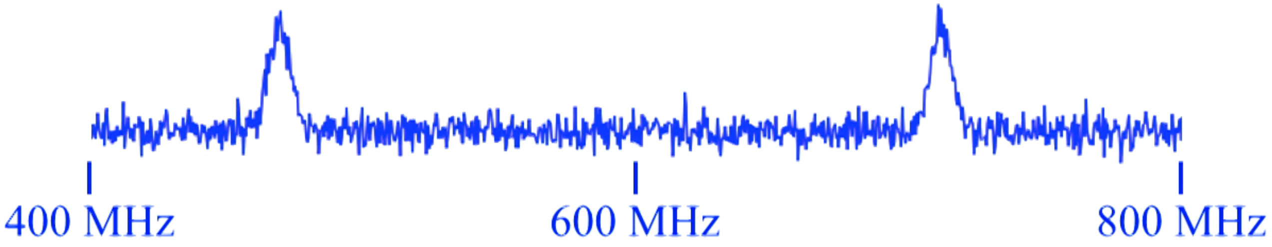
Outline

- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

Cosmology forecasts

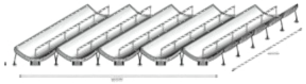
CHIME constrains cosmology by making a 3D map of neutral hydrogen on cosmological scales, via the 21-cm spectral line.

First consider the frequency spectrum of a single line-of-sight. Can be interpreted as a 1D noisy map, in the radial direction.

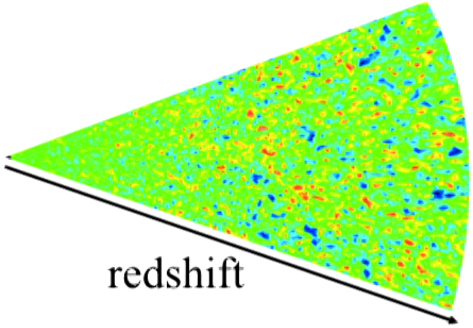


Cosmology forecasts

CHIME measures a 1D spectrum at each 2D sky location (θ, ϕ). Get a 3D map of neutral hydrogen in the universe. (Individual galaxies not resolved, but many Fourier modes are measured.)

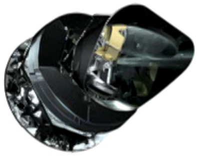


CHIME

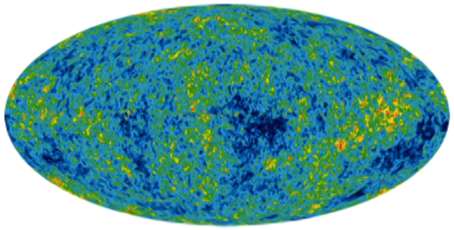


neutral hydrogen density

Conceptually similar to CMB, but much more information is potentially available! (many more modes in 3D than 2D)



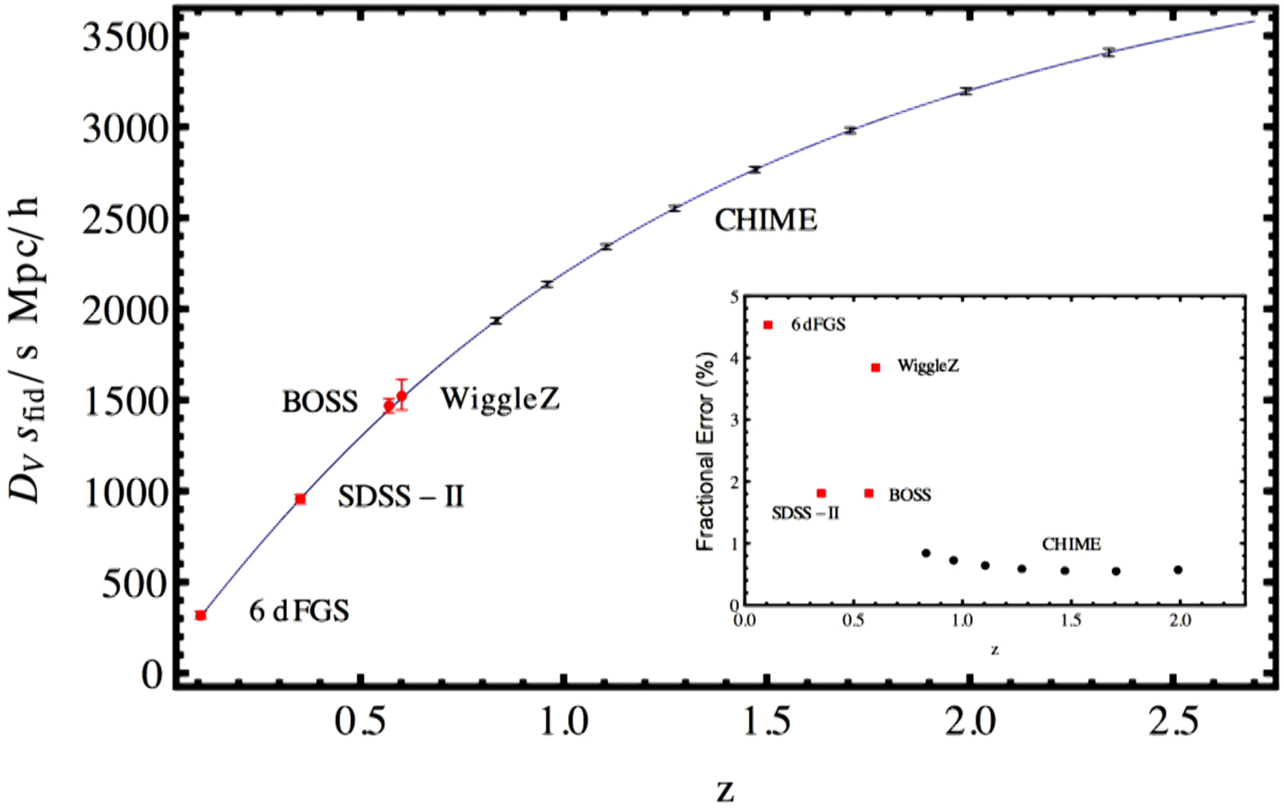
Planck satellite



cosmic microwave background

Expansion history of the universe

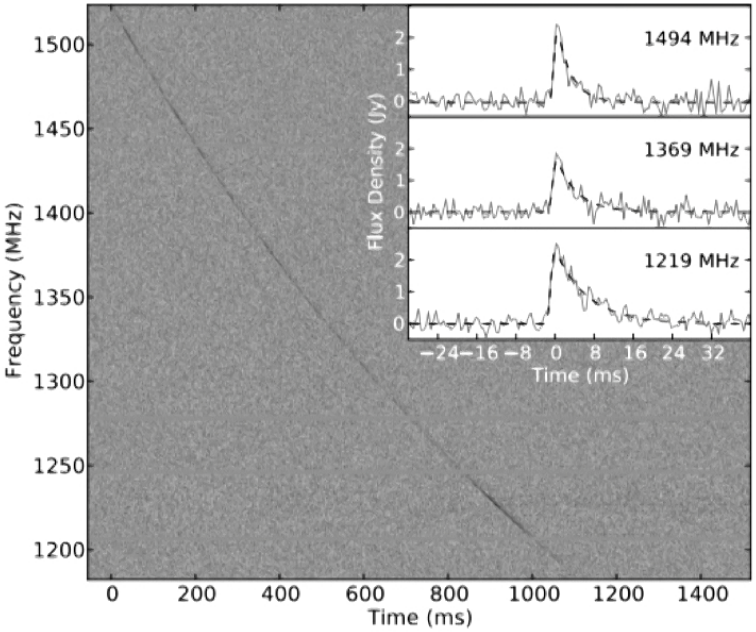
CHIME BAO forecasts are competitive with next-generation surveys (Euclid, LSST, etc.) and probe the "redshift desert"



Fast radio bursts

Fast radio bursts (FRB's): an astrophysical mystery!

Very occasionally, a bright, short (1 ms) pulse of radio emission is observed from an ultra-luminous source at a cosmological distance.

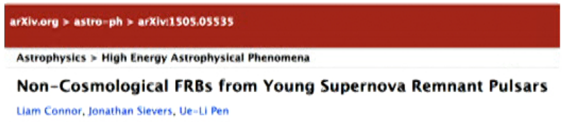


Fast radio bursts

Fast radio bursts (FRB's): an astrophysical mystery!

Very occasionally, a bright, short (1 ms) pulse of radio emission is observed from an ultra-luminous source at a cosmological distance.

The physical mechanism is unknown, and this has become one of the most interesting puzzles in astrophysics!

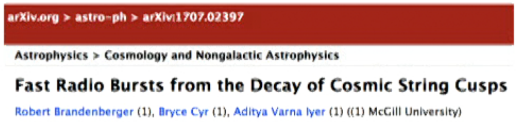


arXiv.org > astro-ph > arXiv:1505.05535

Astrophysics > High Energy Astrophysical Phenomena

Non-Cosmological FRBs from Young Supernova Remnant Pulsars

Liam Connor, Jonathan Sievers, Ue-Li Pen

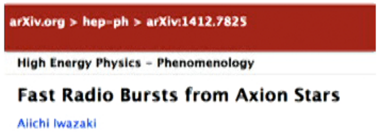


arXiv.org > astro-ph > arXiv:1707.02397

Astrophysics > Cosmology and Nongalactic Astrophysics

Fast Radio Bursts from the Decay of Cosmic String Cusps

Robert Brandenberger (1), Bryce Cyr (1), Aditya Varna Iyer (1) ((1) McGill University)

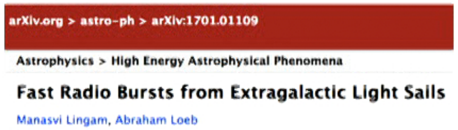


arXiv.org > hep-ph > arXiv:1412.7825

High Energy Physics - Phenomenology

Fast Radio Bursts from Axion Stars

Aiichi Iwazaki



arXiv.org > astro-ph > arXiv:1701.01109

Astrophysics > High Energy Astrophysical Phenomena

Fast Radio Bursts from Extragalactic Light Sails

Manasvi Lingam, Abraham Loeb

arXiv.org > astro-ph > arXiv:1703.00394

Astrophysics > High Energy Astrophysical Phenomena

Giant Primeval Magnetic Dipoles

Christopher Thompson (CITA)

arXiv.org > astro-ph > arXiv:1604.04909

Astrophysics > High Energy Astrophysical Phenomena

The Impact of a Supernova Remnant on Fast Radio Bursts

Anthony L. Piro (Carnegie Observatories)

arXiv.org > astro-ph > arXiv:1511.02870

Astrophysics > High Energy Astrophysical Phenomena

Fast Radio Bursts and Radio Transients from Black Hole Batteries

Chiara M. F. Mingarelli, Janna Levin, T. Joseph W. Lazio

arXiv.org > astro-ph > arXiv:1409.5516

Astrophysics > High Energy Astrophysical Phenomena

Implications of fast radio bursts for superconducting cosmic strings

Yun-Wei Yu, Kwong-Sang Cheng, Gary Shiu, Henry Tye

Fast radio bursts

Fast radio bursts (FRB's): an astrophysical mystery!

Very occasionally, a bright, short (1 ms) pulse of radio emission is observed from an ultra-luminous source at a cosmological distance.

The physical mechanism is unknown, and this has become one of the most interesting puzzles in astrophysics!

Currently statistics-limited: only ~30 FRB's have been observed since the initial discovery in 2007.

CHIME forecasted event rate is…

Fast radio bursts

Fast radio bursts (FRB's): an astrophysical mystery!

Very occasionally, a bright, short (1 ms) pulse of radio emission is observed from an ultra-luminous source at a cosmological distance.

The physical mechanism is unknown, and this has become one of the most interesting puzzles in astrophysics!

Currently statistics-limited: only ~30 FRB's have been observed since the initial discovery in 2007.

CHIME forecasted event rate is… 10 per day!!

Searching for new pulsars

Over the last 50 years, pulsar science has been an exceptionally fertile area of astronomy.

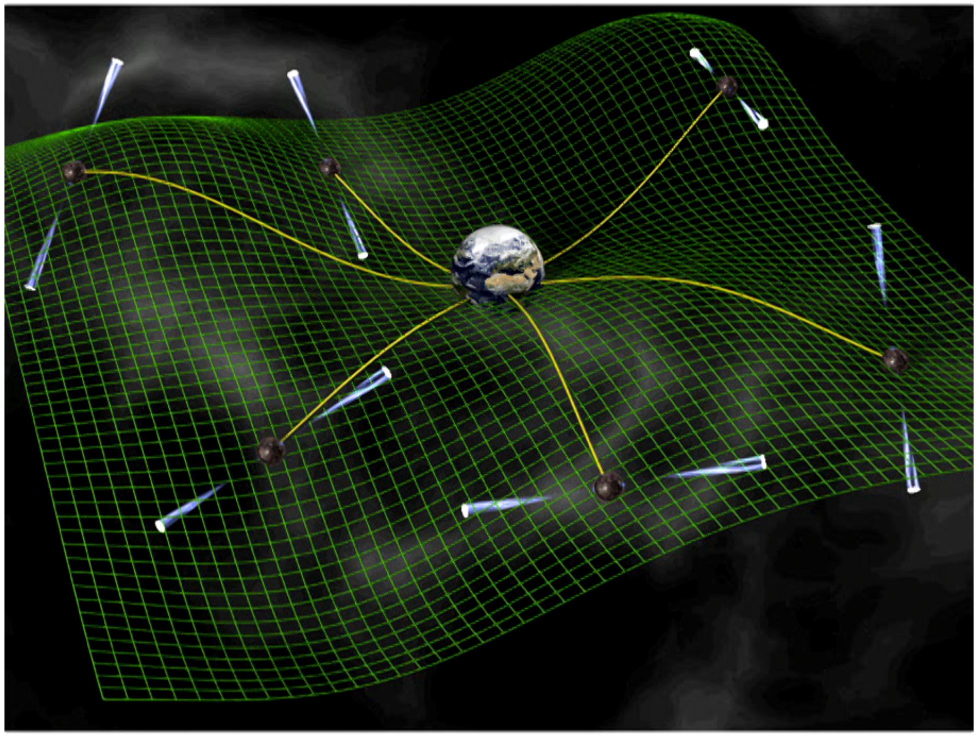
- First observational evidence of gravitational waves
- First detection of extrasolar planets
- Exceptionally precise tests of GR
- New astrophysics (e.g. magnetars)
- Many astronomically interesting "oddball" systems

Around 2000 pulsars have been found, but the total population is predicted to be $\sim 10^5$. Many new discoveries to be made!

We estimate that an optimal search of 30 days of CHIME data would find ~4500 new pulsars!

Timing known pulsars

- Daily timing of known pulsars can contribute to global efforts to detect gravity waves with ~light-year wavelengths.



Timing known pulsars

- Daily timing of known pulsars can contribute to global efforts to detect gravity waves with ~light-year wavelengths.
- CHIME observes at low frequencies (400-800 MHz) where ISM propagation effects can broaden pulses significantly.
- Not ideal for pulsar timing on its own, but when combined with higher-frequency measurements, can improve timing solutions by pinning down ISM-related nuisance parameters
- CHIME will be part of the NANOGRAV pulsar timing collaboration, and we estimate that CHIME can improve timings for most NANOGRAV pulsars by a factor ~2.

CHIME forecasts: summary

- Due to its high mapping speed, CHIME has amazing forecasts in multiple areas (cosmology, fast radio bursts, pulsars).
- CHIME is relatively inexpensive, and any one of these forecasts would fully justify a larger project.
- The CHIME design is highly scalable! A telescope with ~10 or ~100 times the mapping speed could be built in the near future.
- Too good to be true?

Outline

- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

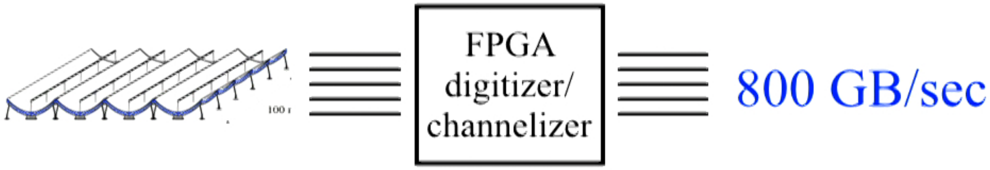
The challenge

	A	N_{beams}	$M/(10^5\ \mathrm{m}^2)$
Parkes 64m	3200 m^2	13	0.41
Green Bank 100m	7850 m^2	7	0.55
Arecibo 300m	70000 m^2	7	4.9
FAST 500m	200000 m^2	19	38
CHIME	6400 m^2	1024	66

In principle, sensitivity is roughly proportional to mapping speed M, but data volume is proportional to N_{beams}.

What we have really done is move difficulty from hardware to software.

CHIME computing

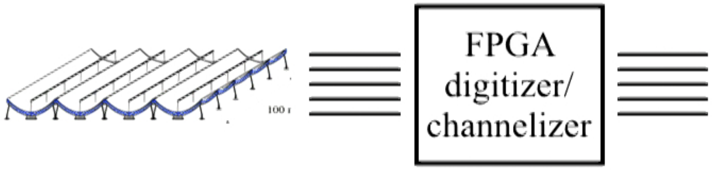


FPGA digitizer/ channelizer

800 GB/sec

- Raw data rate is 800 GB/s = 70 PB/day

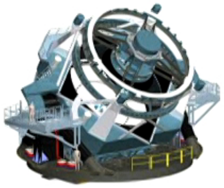
CHIME computing



FPGA digitizer/ channelizer

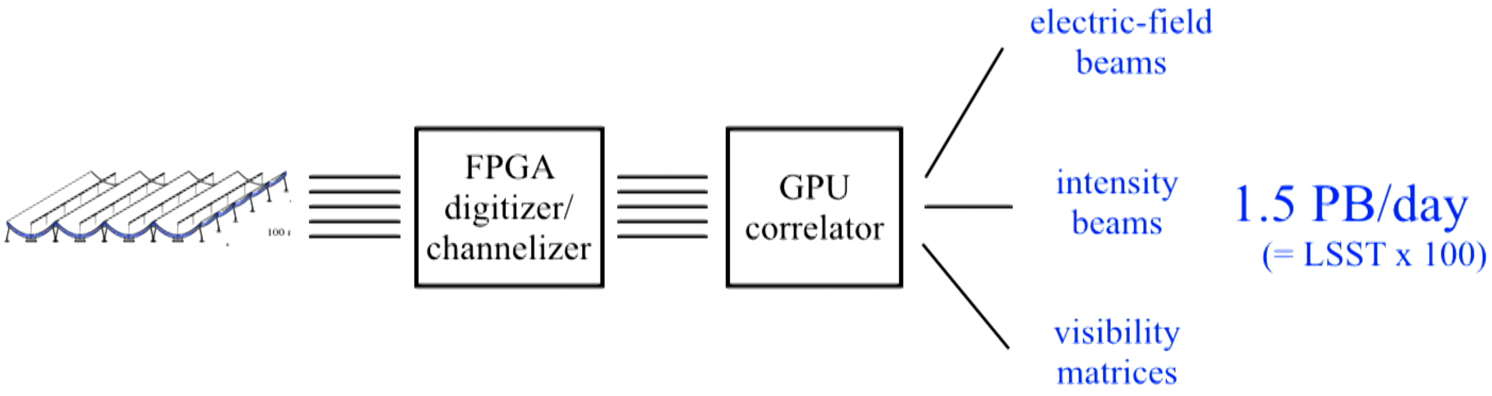
800 GB/sec
(= LSST x 5000)

- Raw data rate is 800 GB/s = 70 PB/day
  = 5000 LSST telescopes!



LSST: 15 TB/day

CHIME computing



FPGA digitizer/ channelizer

GPU correlator

electric-field beams

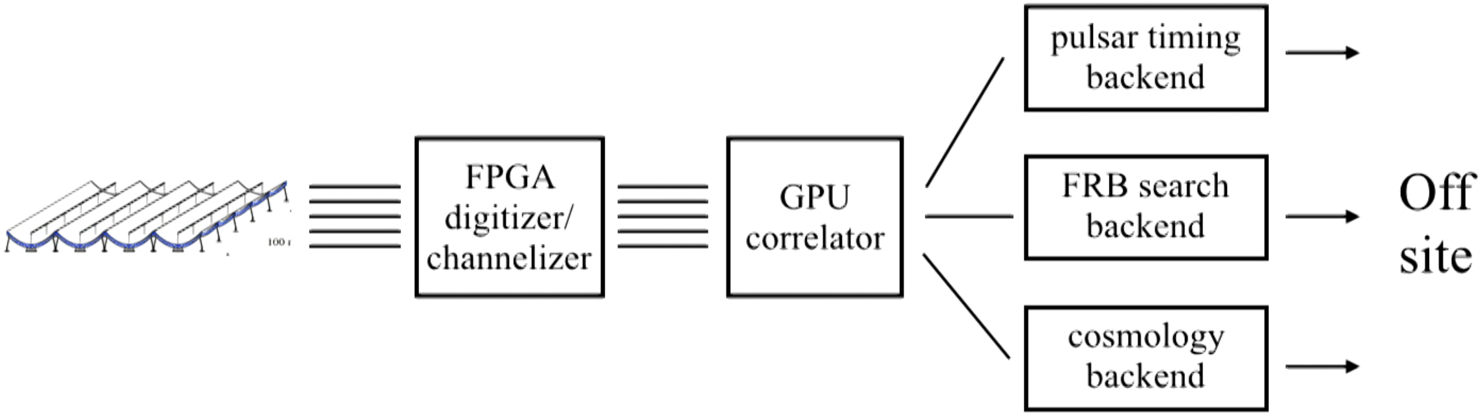
intensity beams

visibility matrices

1.5 PB/day (= LSST x 100)

- Raw data rate is 800 GB/s = 70 PB/day
- 1024-GPU(!) correlator, output data rate is 1.5 PB/day
  - electric-field beams: 10^9 samples/sec at 10 sky locations
  - intensity beams: 10^7 samples/sec at 1024 sky locations
  - visibility matrices: 16GB covariance matrix every 10 seconds

CHIME computing



FPGA digitizer/ channelizer

GPU correlator

pulsar timing backend

FRB search backend

cosmology backend

Off site

- Raw data rate is 800 GB/s = 70 PB/day
- 1024-GPU(!) correlator, output data rate is 1.5 PB/day
  - electric-field beams: 10^9 samples/sec at 10 sky locations
  - intensity beams: 10^7 samples/sec at 1024 sky locations
  - visibility matrices: 16GB covariance matrix every 10 seconds
- Purpose-built backends (e.g. FRB backend is 2560-core cluster!)

Summarizing, the CHIME hardware achieves unprecedented sensitivity at low cost, but requires very hard computing problems to be solved. Subjectively ordered from easiest to hardest:

1. Real-time FRB detection
2. Real-time RFI removal (RFI = "radiofrequency interference", i.e. human-made radio transmissions)
3. Real-time pulsar search
4. Cosmology: separating Galactic foregrounds (synchrotron emission) from cosmological signal (21-cm line emission)

The rest of the talk will be a "status update" on these problems.

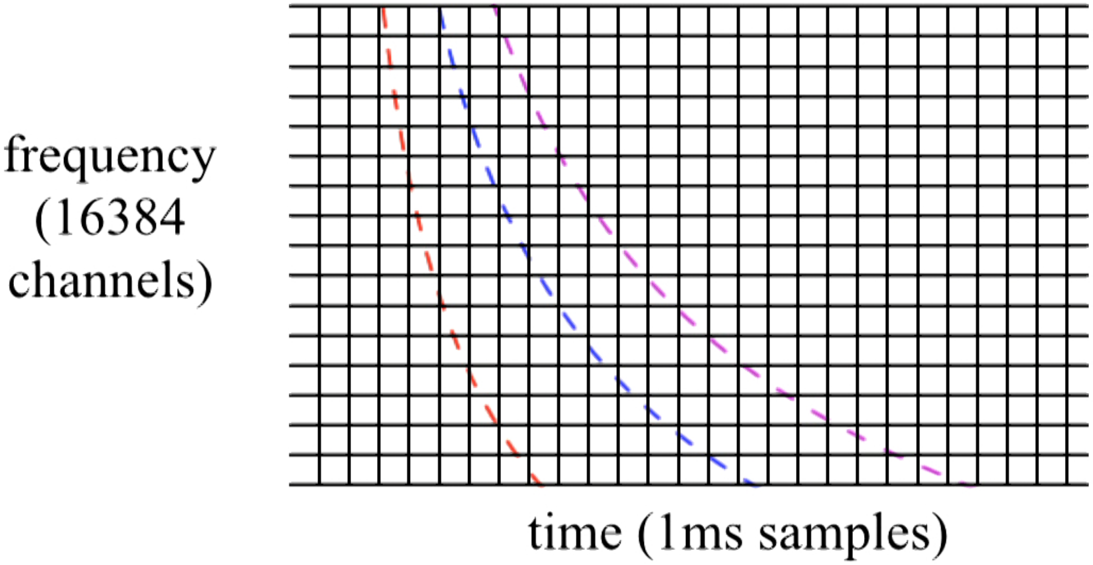
Outline

- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

The FRB search problem

Setting up the problem. The FRB backend incrementally receives a 2D array with (time, frequency) axes. We want to sum over all “tracks” with the shape shown.

We use a recursive tree algorithm, described in the next few slides.

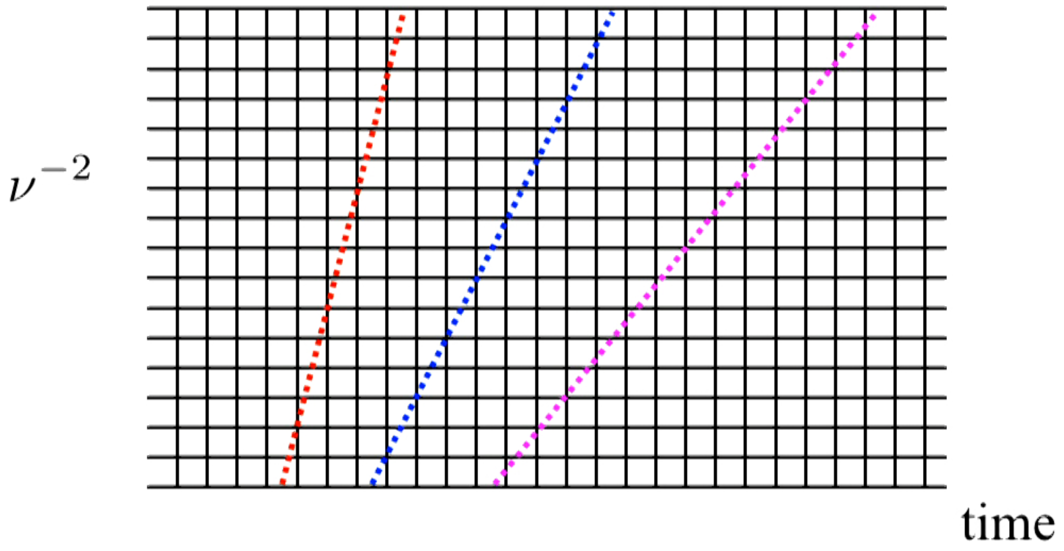


x1024 beams!!

Tree dedispersion: the basic idea

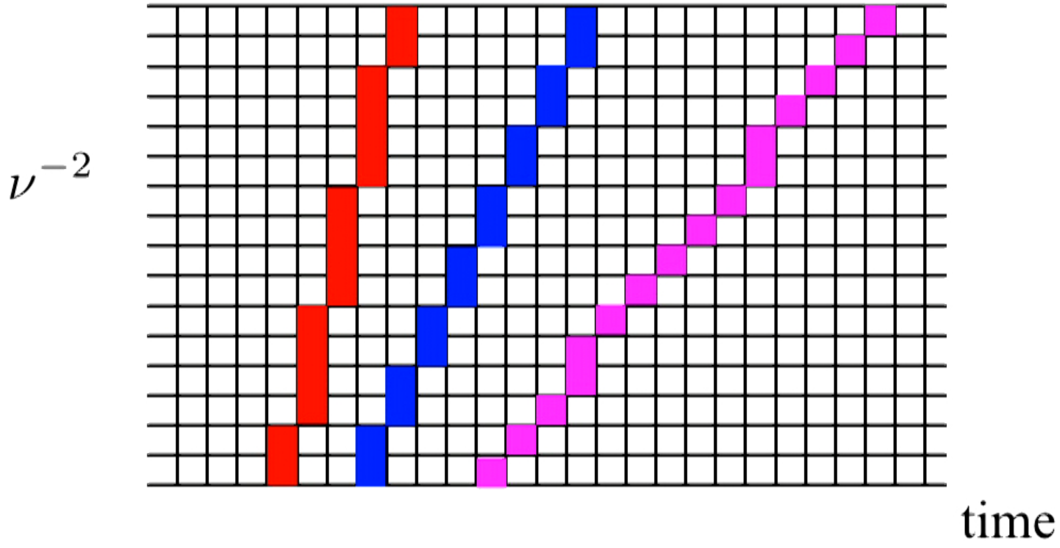
Regrid the input array so that the y-axis corresponds to ν^{-2}, rather than frequency ν.

Then an FRB looks like a straight line. Need a fast algorithm for summing array elements over all straight lines.



Tree dedispersion: the basic idea

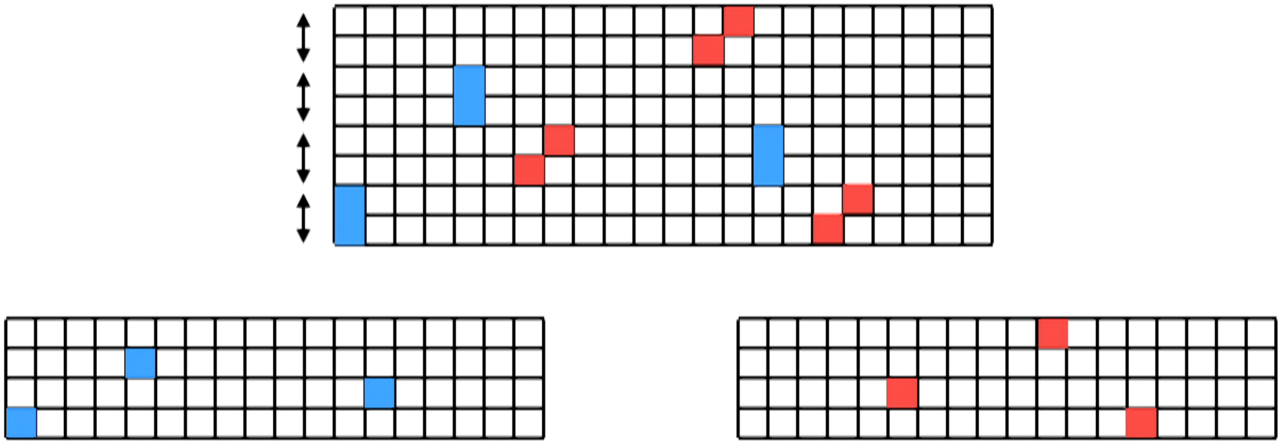
Tree dedispersion will approximate each straight-line track by a jagged sum of samples. The sums are built up recursively as explained in the next few slides.



Tree dedispersion: the basic idea

First iteration: group channels in pairs. Within each pair, we form all "vertical" sums (blue) and "diagonal" sums (red).

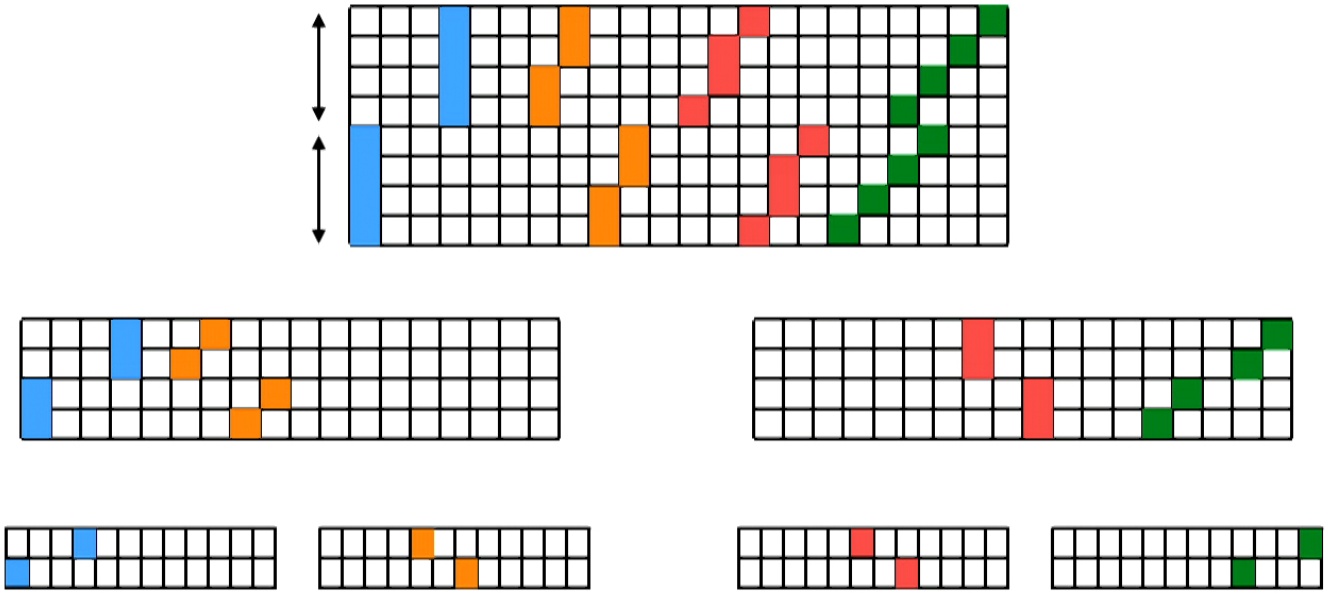
Output is two arrays, each half the size of the input array.



Tree dedispersion: the basic idea

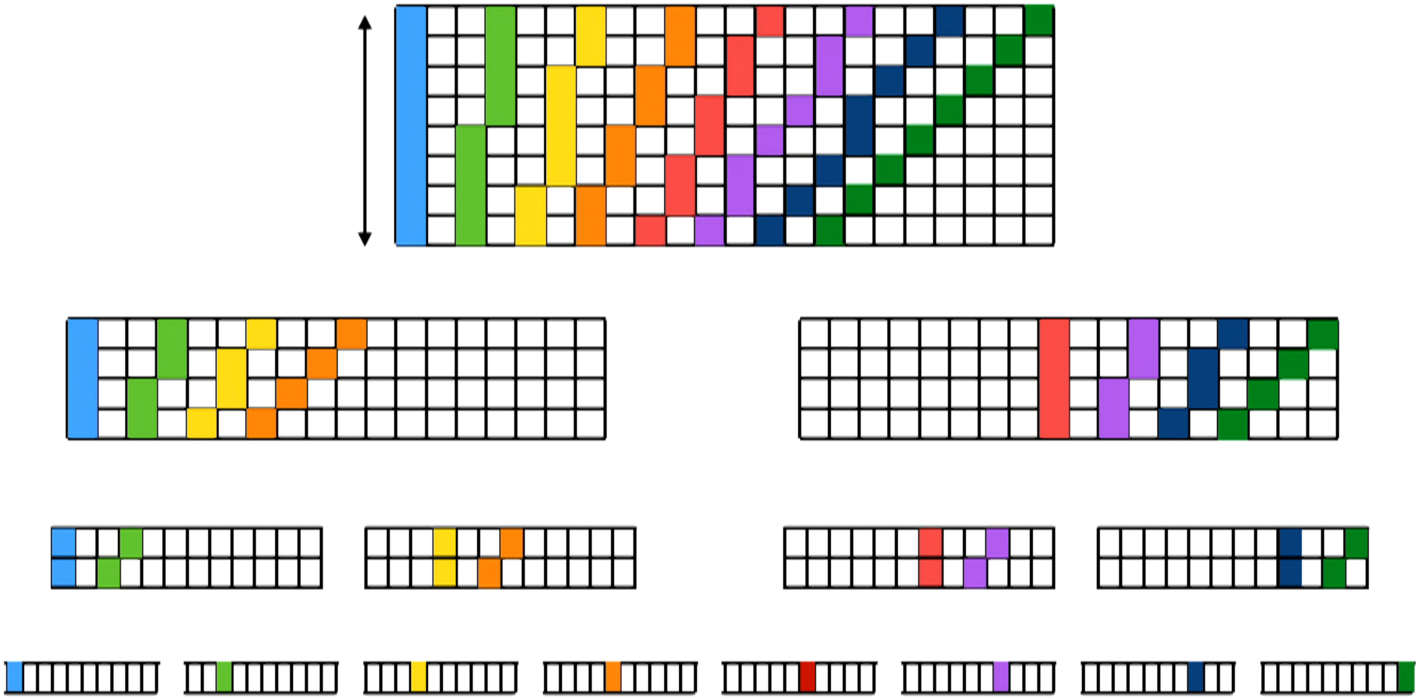
Second iteration: sum pairs into "pairs of pairs".

Frequency channels have now been merged in quadruples. Within each quadruple, there are four possible sums.



Tree dedispersion: the basic idea

Last iteration: all channels summed.



Tree vs direct search

This recursive tree algorithm has computational cost O(TF log F)

T = number of time samples
F = number of frequency channels

In comparison, direct summation would have cost $O(TF^2)$.

For CHIME (F = 16384), tree dedispersion would appear to be ~1000 times faster! However, direct search is more widely used, for two reasons:

- Tree dedispersion tends to badly underperform in practice due to memory bandwidth bottlenecks.
- Tree dedispersion makes approximations, and it is not clear whether it is statistically optimal.

Outline

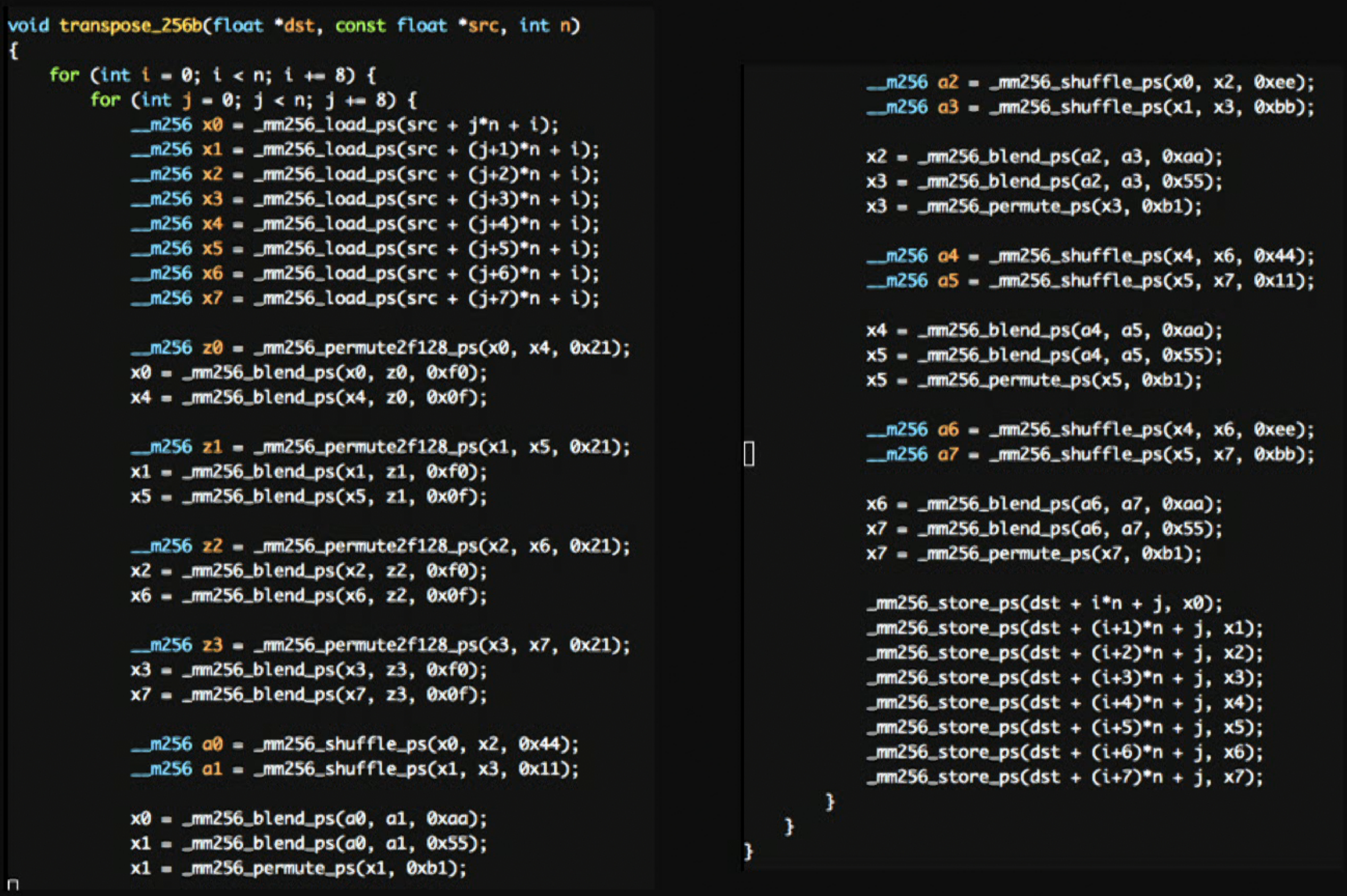
- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

```c
void transpose(float *dst, const float *src, int n)
{
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            dst[i*n+j] = src[j*n+i];
}
```

4 times faster!

```
void transpose_256b(float *dst, const float *src, int n)
{
    for (int i = 0; i < n; i += 8) {
        for (int j = 0; j < n; j += 8) {
            __m256 x0 = _mm256_load_ps(src + j*n + i);
            __m256 x1 = _mm256_load_ps(src + (j+1)*n + i);
            __m256 x2 = _mm256_load_ps(src + (j+2)*n + i);
            __m256 x3 = _mm256_load_ps(src + (j+3)*n + i);
            __m256 x4 = _mm256_load_ps(src + (j+4)*n + i);
            __m256 x5 = _mm256_load_ps(src + (j+5)*n + i);
            __m256 x6 = _mm256_load_ps(src + (j+6)*n + i);
            __m256 x7 = _mm256_load_ps(src + (j+7)*n + i);

            __m256 z0 = _mm256_permute2f128_ps(x0, x4, 0x21);
            x0 = _mm256_blend_ps(x0, z0, 0xf0);
            x4 = _mm256_blend_ps(x4, z0, 0x0f);

            __m256 z1 = _mm256_permute2f128_ps(x1, x5, 0x21);
            x1 = _mm256_blend_ps(x1, z1, 0xf0);
            x5 = _mm256_blend_ps(x5, z1, 0x0f);

            __m256 z2 = _mm256_permute2f128_ps(x2, x6, 0x21);
            x2 = _mm256_blend_ps(x2, z2, 0xf0);
            x6 = _mm256_blend_ps(x6, z2, 0x0f);

            __m256 z3 = _mm256_permute2f128_ps(x3, x7, 0x21);
            x3 = _mm256_blend_ps(x3, z3, 0xf0);
            x7 = _mm256_blend_ps(x7, z3, 0x0f);

            __m256 a0 = _mm256_shuffle_ps(x0, x2, 0x44);
            __m256 a1 = _mm256_shuffle_ps(x1, x3, 0x11);

            x0 = _mm256_blend_ps(a0, a1, 0xaa);
            x1 = _mm256_blend_ps(a0, a1, 0x55);
            x1 = _mm256_permute_ps(x1, 0xb1);

            __m256 a2 = _mm256_shuffle_ps(x0, x2, 0xee);
            __m256 a3 = _mm256_shuffle_ps(x1, x3, 0xbb);

            x2 = _mm256_blend_ps(a2, a3, 0xaa);
            x3 = _mm256_blend_ps(a2, a3, 0x55);
            x3 = _mm256_permute_ps(x3, 0xb1);

            __m256 a4 = _mm256_shuffle_ps(x4, x6, 0x44);
            __m256 a5 = _mm256_shuffle_ps(x5, x7, 0x11);

            x4 = _mm256_blend_ps(a4, a5, 0xaa);
            x5 = _mm256_blend_ps(a4, a5, 0x55);
            x5 = _mm256_permute_ps(x5, 0xb1);

            __m256 a6 = _mm256_shuffle_ps(x4, x6, 0xee);
            __m256 a7 = _mm256_shuffle_ps(x5, x7, 0xbb);

            x6 = _mm256_blend_ps(a6, a7, 0xaa);
            x7 = _mm256_blend_ps(a6, a7, 0x55);
            x7 = _mm256_permute_ps(x7, 0xb1);

            _mm256_store_ps(dst + i*n + j, x0);
            _mm256_store_ps(dst + (i+1)*n + j, x1);
            _mm256_store_ps(dst + (i+2)*n + j, x2);
            _mm256_store_ps(dst + (i+3)*n + j, x3);
            _mm256_store_ps(dst + (i+4)*n + j, x4);
            _mm256_store_ps(dst + (i+5)*n + j, x5);
            _mm256_store_ps(dst + (i+6)*n + j, x6);
            _mm256_store_ps(dst + (i+7)*n + j, x7);
        }
    }
}
```

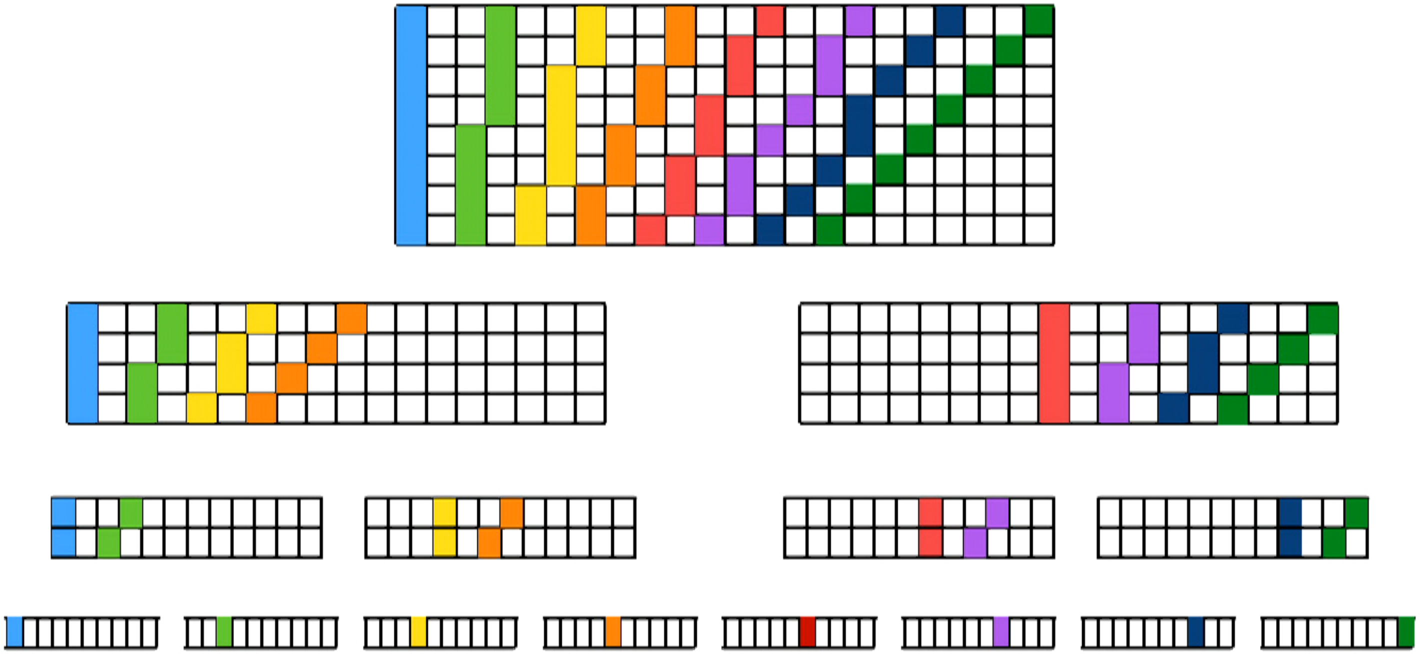
4 times faster!



```
void transpose_256b(float *dst, const float *src, int n)
{
    for (int i = 0; i < n; i += 8) {
        for (int j = 0; j < n; j += 8) {
            __m256 x0 = _mm256_load_ps(src + j*n + i);
            __m256 x1 = _mm256_load_ps(src + (j+1)*n + i);
            __m256 x2 = _mm256_load_ps(src + (j+2)*n + i);
            __m256 x3 = _mm256_load_ps(src + (j+3)*n + i);
            __m256 x4 = _mm256_load_ps(src + (j+4)*n + i);
            __m256 x5 = _mm256_load_ps(src + (j+5)*n + i);
            __m256 x6 = _mm256_load_ps(src + (j+6)*n + i);
            __m256 x7 = _mm256_load_ps(src + (j+7)*n + i);

            __m256 z0 = _mm256_permute2f128_ps(x0, x4, 0x21);
            x0 = _mm256_blend_ps(x0, z0, 0xf0);
            x4 = _mm256_blend_ps(x4, z0, 0x0f);

            __m256 z1 = _mm256_permute2f128_ps(x1, x5, 0x21);
            x1 = _mm256_blend_ps(x1, z1, 0xf0);
            x5 = _mm256_blend_ps(x5, z1, 0x0f);

            __m256 z2 = _mm256_permute2f128_ps(x2, x6, 0x21);
            x2 = _mm256_blend_ps(x2, z2, 0xf0);
            x6 = _mm256_blend_ps(x6, z2, 0x0f);

            __m256 z3 = _mm256_permute2f128_ps(x3, x7, 0x21);
            x3 = _mm256_blend_ps(x3, z3, 0xf0);
            x7 = _mm256_blend_ps(x7, z3, 0x0f);

            __m256 a0 = _mm256_shuffle_ps(x0, x2, 0x44);
            __m256 a1 = _mm256_shuffle_ps(x1, x3, 0x11);

            x0 = _mm256_blend_ps(a0, a1, 0xaa);
            x1 = _mm256_blend_ps(a0, a1, 0x55);
            x1 = _mm256_permute_ps(x1, 0xb1);

            __m256 a2 = _mm256_shuffle_ps(x0, x2, 0xee);
            __m256 a3 = _mm256_shuffle_ps(x1, x3, 0xbb);

            x2 = _mm256_blend_ps(a2, a3, 0xaa);
            x3 = _mm256_blend_ps(a2, a3, 0x55);
            x3 = _mm256_permute_ps(x3, 0xb1);

            __m256 a4 = _mm256_shuffle_ps(x4, x6, 0x44);
            __m256 a5 = _mm256_shuffle_ps(x5, x7, 0x11);

            x4 = _mm256_blend_ps(a4, a5, 0xaa);
            x5 = _mm256_blend_ps(a4, a5, 0x55);
            x5 = _mm256_permute_ps(x5, 0xb1);

            __m256 a6 = _mm256_shuffle_ps(x4, x6, 0xee);
            __m256 a7 = _mm256_shuffle_ps(x5, x7, 0xbb);

            x6 = _mm256_blend_ps(a6, a7, 0xaa);
            x7 = _mm256_blend_ps(a6, a7, 0x55);
            x7 = _mm256_permute_ps(x7, 0xb1);

            _mm256_store_ps(dst + i*n + j, x0);
            _mm256_store_ps(dst + (i+1)*n + j, x1);
            _mm256_store_ps(dst + (i+2)*n + j, x2);
            _mm256_store_ps(dst + (i+3)*n + j, x3);
            _mm256_store_ps(dst + (i+4)*n + j, x4);
            _mm256_store_ps(dst + (i+5)*n + j, x5);
            _mm256_store_ps(dst + (i+6)*n + j, x6);
            _mm256_store_ps(dst + (i+7)*n + j, x7);
        }
    }
}
```

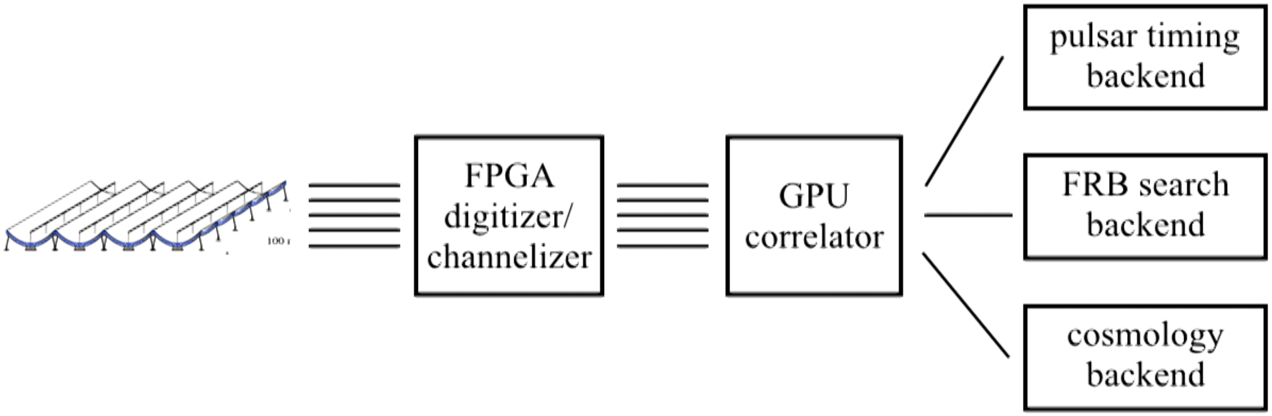
“Blocked” tree dedispersion

- In a straightforward implementation of tree dedispersion, L3 cache bandwidth is the limiting factor.
- Non-obvious fact: tree dedispersion can be “blocked” so that the array goes through L3 cache twice! This gives a huge speedup.



Incremental tree dedispersion

- An interesting property of the blocked algorithm is that it is also incremental.
- Data is processed incrementally in few-second “chunks”. An FRB is detected within a few seconds, even though the dispersion delay of the FRB may be ~minutes.
- Useful for real-time alerts. We also plan to use it internally to CHIME, to trigger dumps of baseband (=electric field) data.



FPGA digitizer/ channelizer

GPU correlator

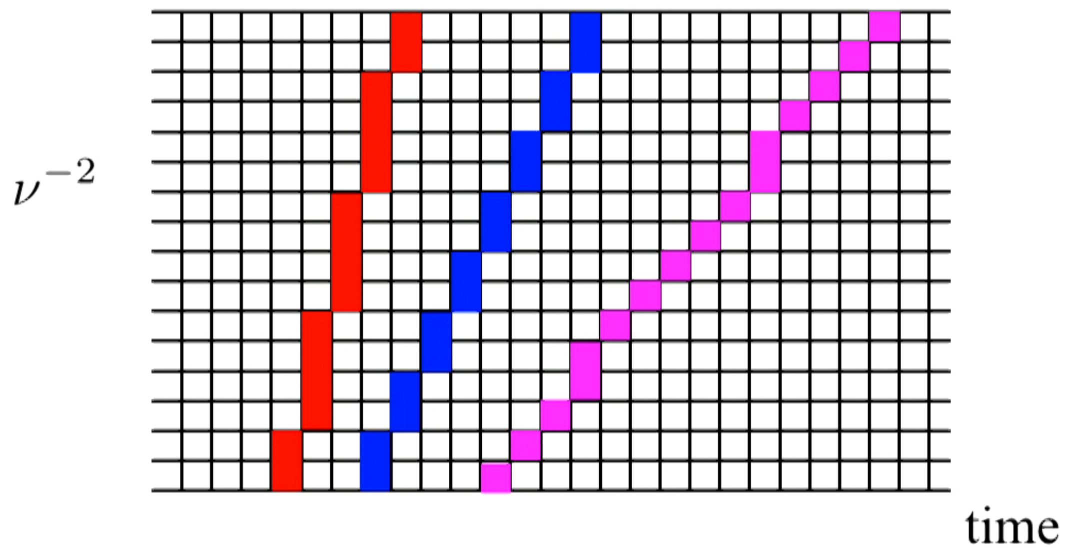
pulsar timing backend

FRB search backend

cosmology backend

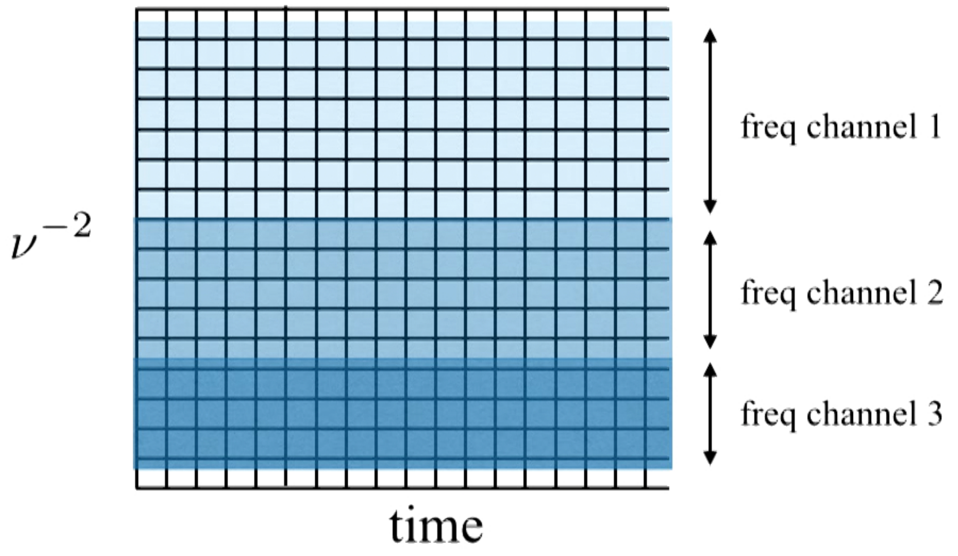
How optimal is tree dedispersion?

In principle, tree dedispersion is not a perfectly optimal search: straight lines in (t, ν^{-2}) plane are approximated by jagged lines.



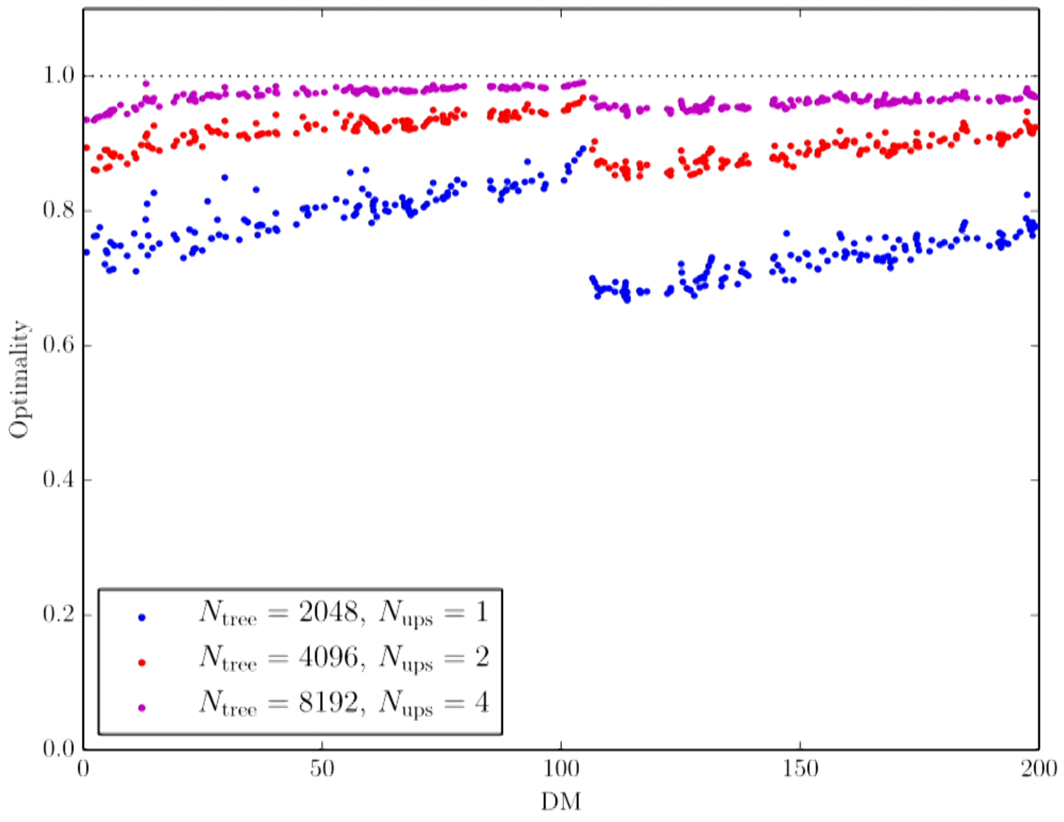
Theorem: tree dedispersion is optimal in the following limit:

- The number of tree channels (N_{tree}) is >> the number of frequency channels (N_{freq}), and regridding of frequency channels to tree channels is done by "smearing".
- The tree is upsampled in time relative to the data (by duplicating samples in the time direction, in the initial regridding step).



Toy example: 1024 frequency channels, search to DM_{max}=200. Each point in the scatterplot is one Monte Carlo simulation.

"Optimality" = ($SNR_{recovered}$ / SNR_{true}).



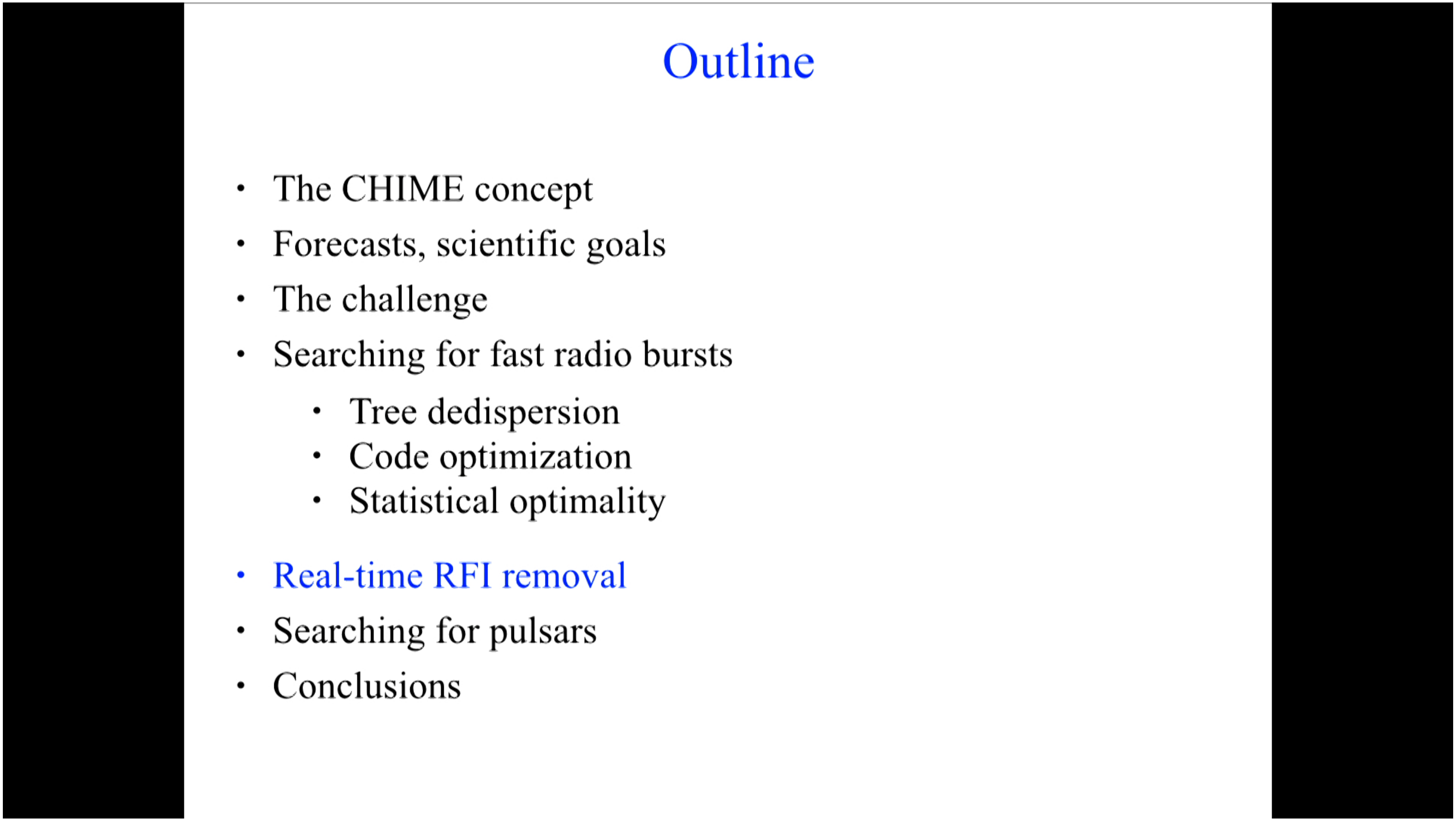
In summary, our FRB search algorithm is:

- provably near-optimal
- real-time, with very low (few-second) latency
- much faster than other search software

How much faster?

- The CHIME FRB search will be the world's largest in data volume, by a factor 1700!
- With our algorithm, we could process the second-largest search (the 36-beam ASKAP survey) in real time with 0.5 CPU cores. This level of performance is necessary for CHIME!

Smith et al, to appear

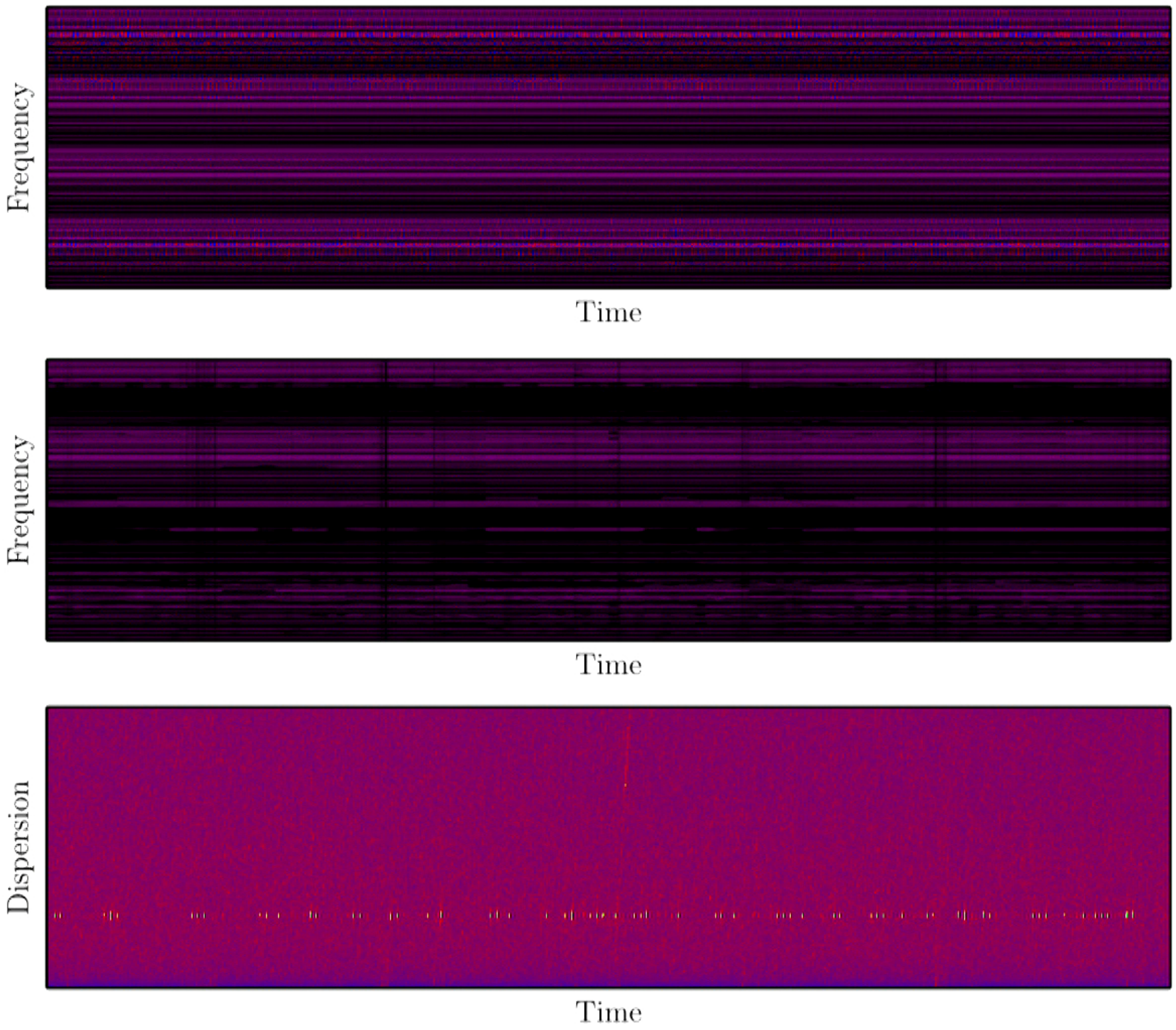


Outline

- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

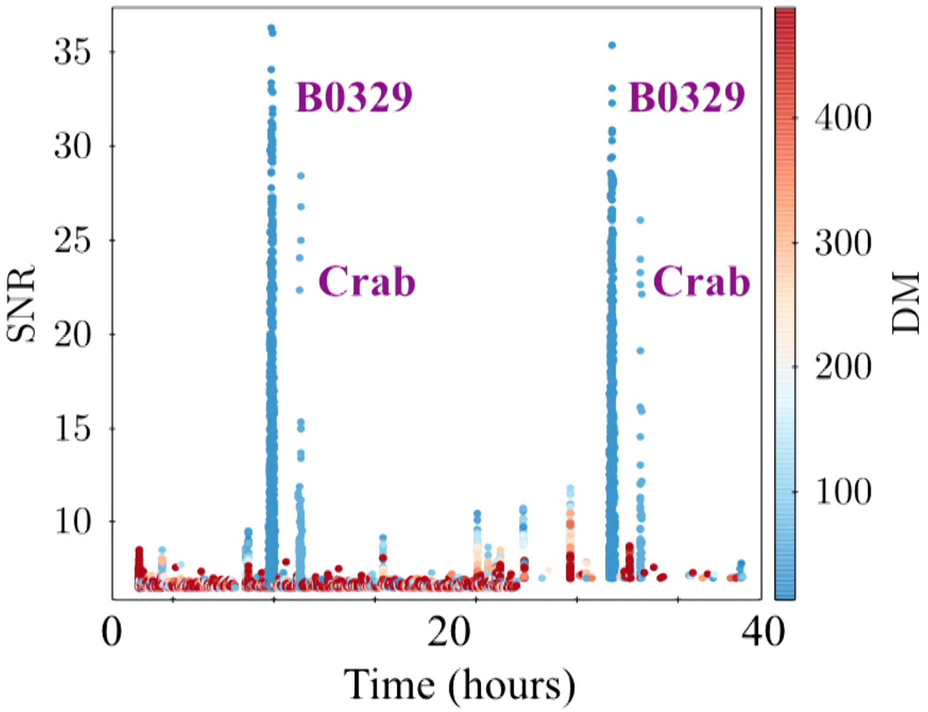
RFI removal

- For an FRB or pulsar search, the largest instrumental effect (by far!) is radiofrequency interference (RFI), i.e. human-made radio transmissions.
- Main tool for mitigating RFI is masking the data in the (time, frequency) plane, before the FRB search.
- Standard RFI removal software packages do not suffice for CHIME:
  - too slow
  - latency too high
  - false positive rate too high (a few false positives per beam per hour = 10^5 events per day!)


Frequency
Time
Frequency
Time
Dispersion
Time

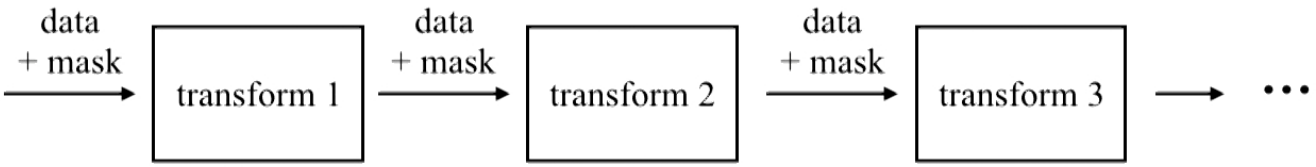
RFI victory plot: pulse significance (in “sigmas”) vs time, in a single-beam, 2-day run in December.

Real-time RFI removal is working extremely well!



Our approach to RFI removal

Represent RFI removal as a sequence of “transforms” which operate on the data + mask.



data + mask → transform 1 → data + mask → transform 2 → data + mask → transform 3 → …

For example:

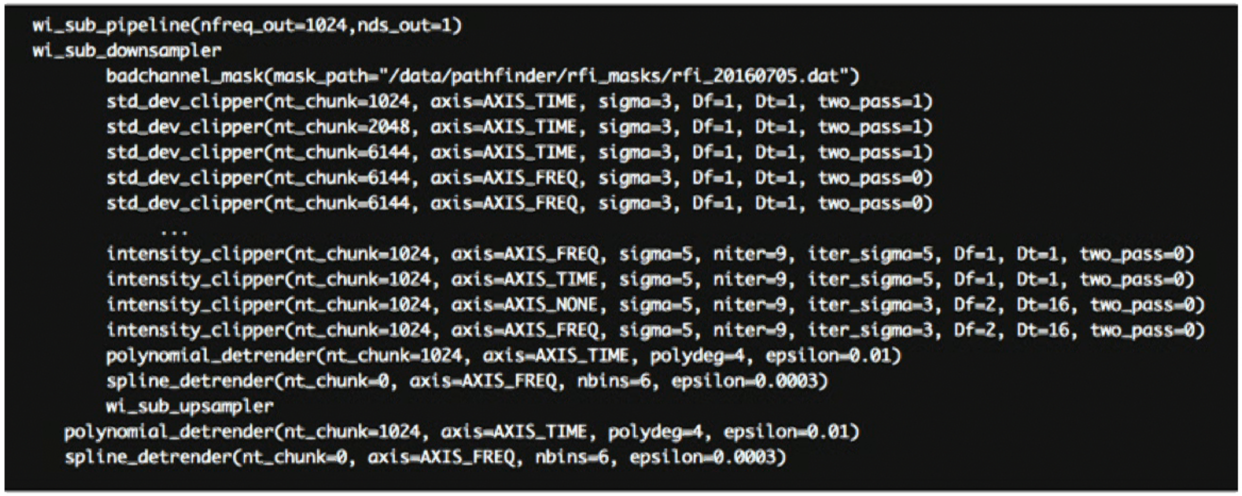
- Clipping based on intensity
- Clipping based on variance of intensity (voltage “kurtosis”)
- Detrending the data in either time or frequency axis
- Upsampling/downsampling the data/mask

Our key transforms are assembly-language-kernelized, but can be chained together and run from high-level languages (python).

Our approach to RFI removal

Current RFI strategy consists of ~100 transforms!

Finding this sequence required a lot of experimentation on archival data (led by Masoud Rafiei-Ravandi).



```
wi_sub_pipeline(nfreq_out=1024,nds_out=1)
wi_sub_downsampler
       badchannel_mask(mask_path="/data/pathfinder/rfi_masks/rfi_20160705.dat")
       std_dev_clipper(nt_chunk=1024, axis=AXIS_TIME, sigma=3, Df=1, Dt=1, two_pass=1)
       std_dev_clipper(nt_chunk=2048, axis=AXIS_TIME, sigma=3, Df=1, Dt=1, two_pass=1)
       std_dev_clipper(nt_chunk=6144, axis=AXIS_TIME, sigma=3, Df=1, Dt=1, two_pass=1)
       std_dev_clipper(nt_chunk=6144, axis=AXIS_FREQ, sigma=3, Df=1, Dt=1, two_pass=0)
       std_dev_clipper(nt_chunk=6144, axis=AXIS_FREQ, sigma=3, Df=1, Dt=1, two_pass=0)
            ...
       intensity_clipper(nt_chunk=1024, axis=AXIS_FREQ, sigma=5, niter=9, iter_sigma=5, Df=1, Dt=1, two_pass=0)
       intensity_clipper(nt_chunk=1024, axis=AXIS_TIME, sigma=5, niter=9, iter_sigma=5, Df=1, Dt=1, two_pass=0)
       intensity_clipper(nt_chunk=1024, axis=AXIS_NONE, sigma=5, niter=9, iter_sigma=3, Df=2, Dt=16, two_pass=0)
       intensity_clipper(nt_chunk=1024, axis=AXIS_FREQ, sigma=5, niter=9, iter_sigma=3, Df=2, Dt=16, two_pass=0)
       polynomial_detrender(nt_chunk=1024, axis=AXIS_TIME, polydeg=4, epsilon=0.01)
       spline_detrender(nt_chunk=0, axis=AXIS_FREQ, nbins=6, epsilon=0.0003)
       wi_sub_upsampler
   polynomial_detrender(nt_chunk=1024, axis=AXIS_TIME, polydeg=4, epsilon=0.01)
   spline_detrender(nt_chunk=0, axis=AXIS_FREQ, nbins=6, epsilon=0.0003)
```

RFI: future plans

Open question: does a general solution to the RFI removal problem exist? Since it requires so much telescope-dependent tuning, every project tends to write new software from scratch!

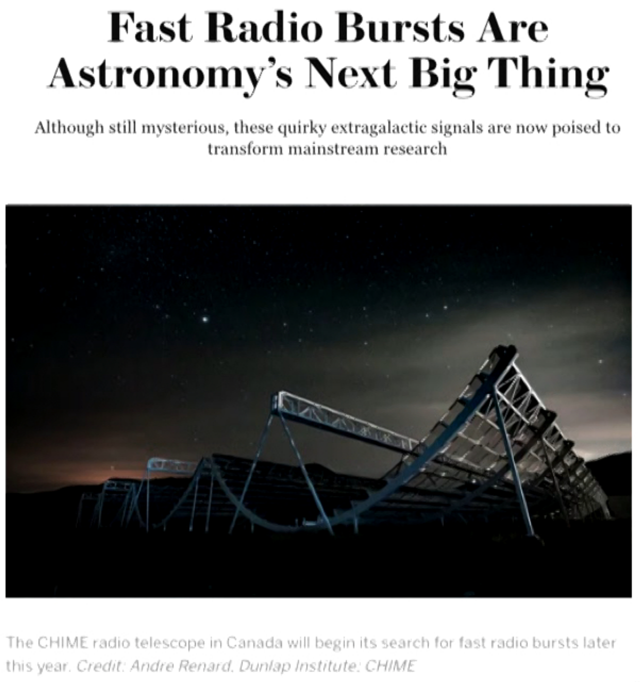
We speculate that the following approach may work:

- Develop a toolkit of "transforms", or low-level building blocks.
- Develop a learning algorithm which can "evolve" a transform chain, given a large training set of archival data.

If this works, it would be of immense practical importance!

In the meantime, CHIME real-time RFI removal is already working extremely well.

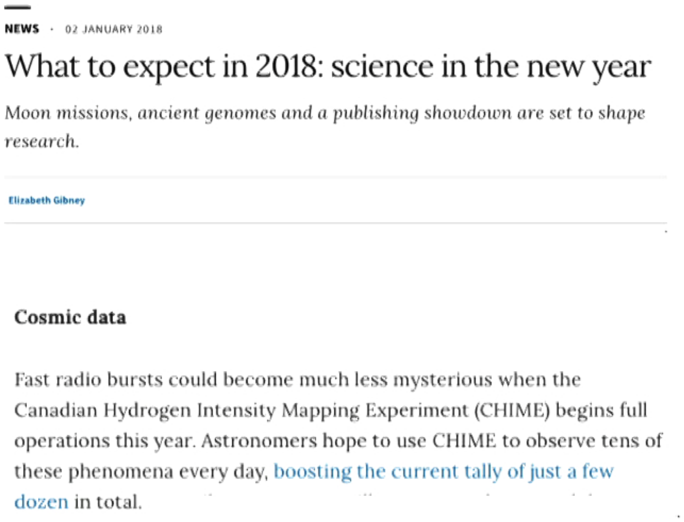
We are currently building a 2560-core cluster to run the CHIME FRB search. Construction should be finished in late May!



Fast Radio Bursts Are Astronomy's Next Big Thing

Although still mysterious, these quirky extragalactic signals are now poised to transform mainstream research

The CHIME radio telescope in Canada will begin its search for fast radio bursts later this year. *Credit: Andre Renard, Dunlap Institute; CHIME*



NEWS · 02 JANUARY 2018

What to expect in 2018: science in the new year

Moon missions, ancient genomes and a publishing showdown are set to shape research.

Elizabeth Gibney

Cosmic data

Fast radio bursts could become much less mysterious when the Canadian Hydrogen Intensity Mapping Experiment (CHIME) begins full operations this year. Astronomers hope to use CHIME to observe tens of these phenomena every day, boosting the current tally of just a few dozen in total.

https://www.scientificamerican.com/article/fast-radio-bursts-are-astronomy-rsquo-s-next-big-thing/

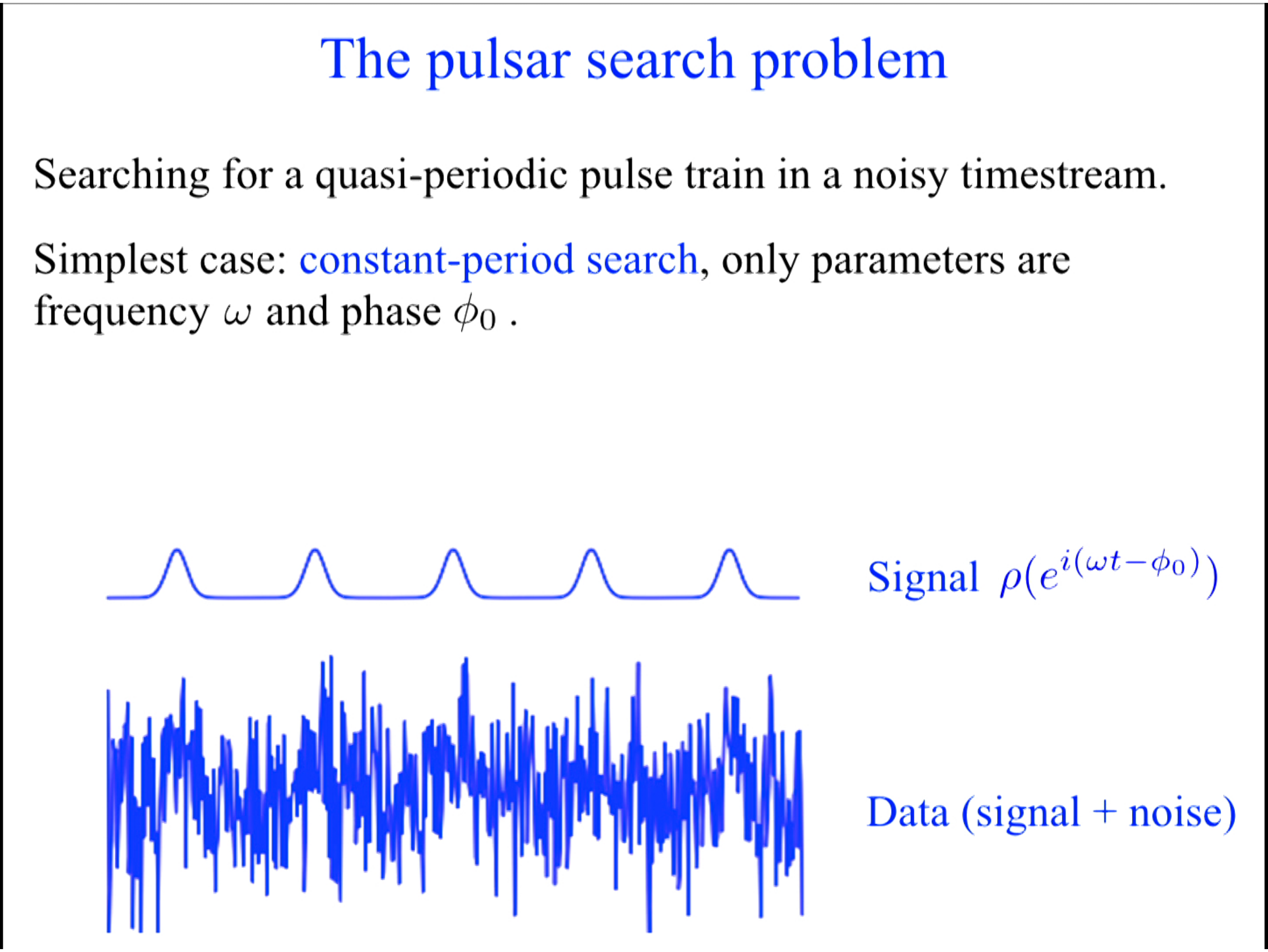
https://www.nature.com/articles/d41586-018-00009-5

Outline

- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

Searching for pulsars with CHIME

- The CHIME pulsar search is where the FRB search was a few years ago: forecasts look amazing, no software written yet, big computational challenges remain.
- Methods paper (KMS, 1610.06831) proposing new algorithms.
- We estimate that an optimal search of 30 days of CHIME data would find ~4500 new pulsars!
- Over the summer, we plan to do a lot of software development and assess whether this target is achievable.
- We plan to add a fourth computing backend to CHIME: a pulsar search backend.



The pulsar search problem

Searching for a quasi-periodic pulse train in a noisy timestream.

Simplest case: constant-period search, only parameters are frequency ω and phase ϕ_0 .

Signal $\rho(e^{i(\omega t - \phi_0)})$

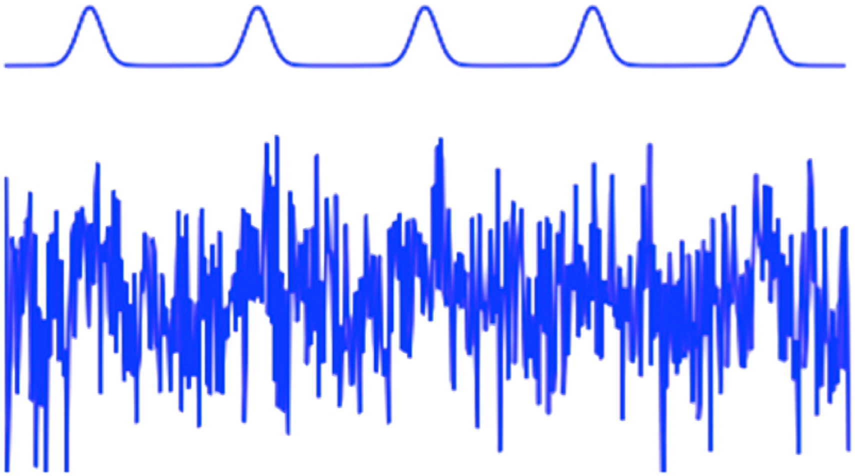
Data (signal + noise)

Constant-period pulsar search

Optimal search algorithm: “trial folding”

- Outer loop over trial parameters (ω, ϕ_0)
- Sum over data samples d_k, weighted by expected signal

$$\hat{\mathcal{E}}(\omega, \phi_0) = \sum_k d_k \int_{(k-1/2)t_s}^{(k+1/2)t_s} dt\, \rho\Big(e^{i(\omega t - \phi_0)}\Big)$$



Constant-period pulsar search

Optimal search algorithm: “trial folding”

- Outer loop over trial parameters (ω, ϕ_0)
- Sum over data samples d_k, weighted by expected signal

$$\hat{\mathcal{E}}(\omega, \phi_0) = \sum_k d_k \int_{(k-1/2)t_s}^{(k+1/2)t_s} dt\, \rho\Big(e^{i(\omega t - \phi_0)}\Big)$$

Looks slow, but there is a fast algorithm (KMS, 1610.06831). Idea: interpret trial folding as an integral transform

$$d(t) \to \hat{\mathcal{E}}(\omega, \phi_0)$$

Is there a fast way to compute it?

$$\hat{\mathcal{E}}(\omega,\phi_0) = \sum_k d_k \int_{(k-1/2)t_s}^{(k+1/2)t_s} dt\, \rho\left(e^{i(\omega t-\phi_0)}\right)$$

definition

$$= \sum_k d_k \int_{(k-1/2)t_s}^{(k+1/2)t_s} dt \sum_n \rho_n e^{in(\omega t-\phi_0)}$$

Fourier transform ρ

$$= \sum_k d_k \sum_n \rho_n \frac{\sin(n\omega t_s/2)}{n\omega t_s/2} e^{in(\omega k t_s-\phi_0)}$$

do integral

$$= \sum_n \rho_n e^{-in\phi_0} \frac{\sin(n\omega t_s/2)}{n\omega t_s/2} \sum_k d_k e^{in\omega k t_s}$$

reverse order
of summation

Fast algorithm:

- Compute $\tilde{d}(m) = \sum_k d_k e^{ikm}$ using FFT algorithm
- Compute $\tilde{E}(\omega,n) = \rho_n \frac{\sin(n\omega t_s/2)}{n\omega t_s/2}\, \tilde{d}(n\omega t_s)$ by interpolation
- Compute $\hat{\mathcal{E}}(\omega,\phi_0) = \sum_n \tilde{E}(\omega,n) e^{-in\phi_0}$ using FFT algorithm

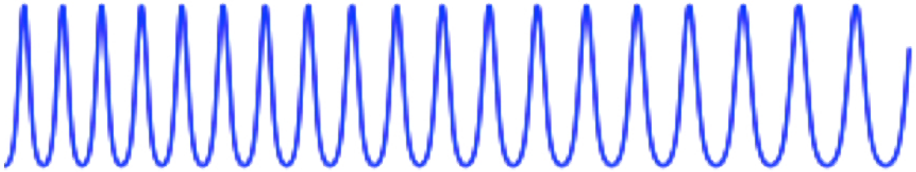
Constant-acceleration pulsar search

Now consider the most common pulsar search: the "constant-acceleration" search with three parameters.

$\alpha =$ acceleration/deceleration

$\omega =$ pulse frequency

$\phi_0 =$ phase



Signal $\rho\left(e^{i(\alpha t^2+\omega t-\phi_0)}\right)$

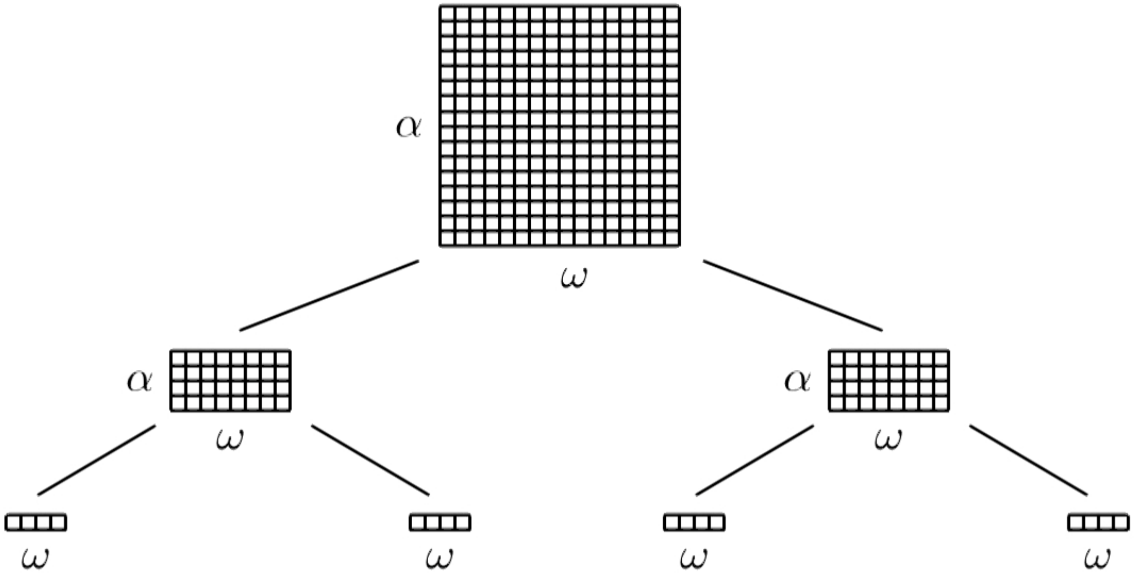
Leads to integral transform:

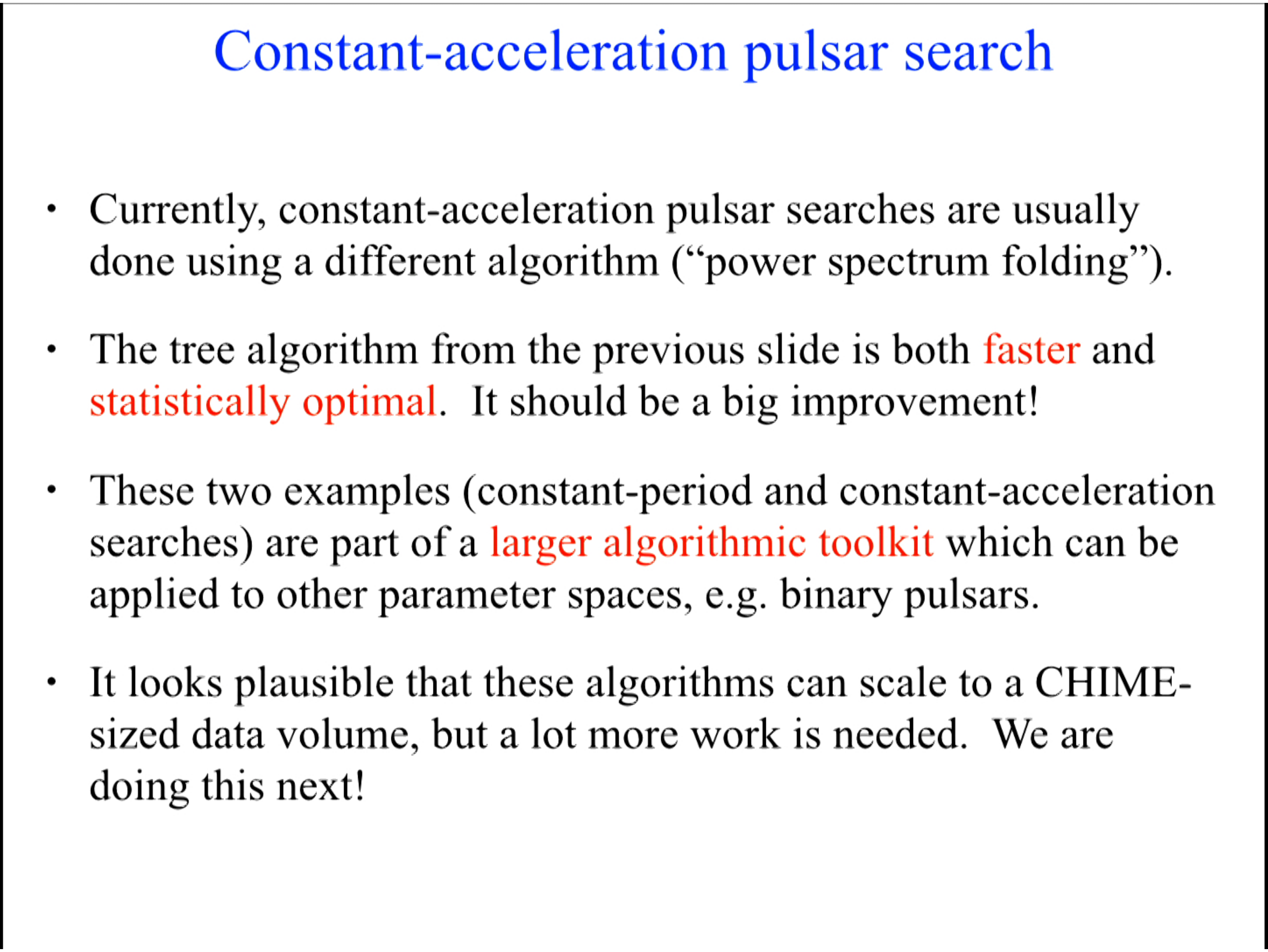
$$\hat{\mathcal{E}}(\alpha,\omega,\phi_0) = \sum_k d_k \int_{(k-1/2)t_s}^{(k+1/2)t_s} dt\, \rho\left(e^{i(\alpha t^2+\omega t-\phi_0)}\right)$$

$$\hat{\mathcal{E}}(\alpha,\omega,\phi_0) = \sum_k d_k \int_{(k-1/2)t_s}^{(k+1/2)t_s} dt\, \rho\Big(e^{i(\alpha t^2+\omega t-\phi_0)}\Big)$$

Can be computed efficiently by a recursive tree algorithm. Divide time interval in two, and compute arrays $(\hat{\mathcal{E}}_1, \hat{\mathcal{E}}_2)$ for the two halves. Then use recursion relation of the form:

$$\hat{\mathcal{E}}(\alpha,\omega,\phi_0) = \hat{\mathcal{E}}_1(\alpha',\omega',\phi_0') + \hat{\mathcal{E}}(\alpha'',\omega'',\phi_0'')$$





Constant-acceleration pulsar search

- Currently, constant-acceleration pulsar searches are usually done using a different algorithm (“power spectrum folding”).
- The tree algorithm from the previous slide is both faster and statistically optimal. It should be a big improvement!
- These two examples (constant-period and constant-acceleration searches) are part of a larger algorithmic toolkit which can be applied to other parameter spaces, e.g. binary pulsars.
- It looks plausible that these algorithms can scale to a CHIME-sized data volume, but a lot more work is needed. We are doing this next!

Outline

- The CHIME concept
- Forecasts, scientific goals
- The challenge
- Searching for fast radio bursts
    - Tree dedispersion
    - Code optimization
    - Statistical optimality
- Real-time RFI removal
- Searching for pulsars
- Conclusions

Conclusions

We are making excellent progress on solving the hard data analysis problems needed for CHIME to fulfill its potential.

1. Real-time FRB detection
   solved!
2. Real-time RFI removal
   solved!
3. Real-time pulsar search
   in progress
4. Cosmology: separating Galactic foregrounds from cosmological signal
   in progress

Conclusions

	A	N_{beams}	$M/(10^5\ \mathrm{m}^2)$
Parkes 64m	$3200\ \mathrm{m}^2$	13	0.41
Green Bank 100m	$7850\ \mathrm{m}^2$	7	0.55
Arecibo 300m	$70000\ \mathrm{m}^2$	7	4.9
FAST 500m	$200000\ \mathrm{m}^2$	19	38
CHIME	$6400\ \mathrm{m}^2$	1024	66

For some problems (e.g. finding FRB's), CHIME will be the most powerful telescope on this list, as far as I can tell.

CHIME is (by far!) the least expensive telescope on the list, and could be scaled up to 10-100 times higher mapping speed.

Conclusions

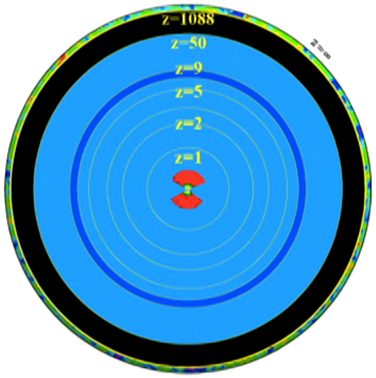
The beginning of an era in radio astronomy: “large N and clever algorithms”?

Example: HIRAX, a South African “sister” project to CHIME

- Array of 1024 dishes (no cylinders)
- 4 times more collecting area than CHIME
- Outrigger telescopes for very high resolution!
- In Southern hemisphere (more pulsars)

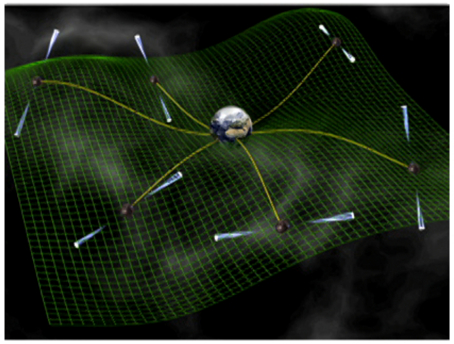
Conclusions

Radio astronomy may be "scaled up" by orders of magnitude in the near future. The discovery space is huge!



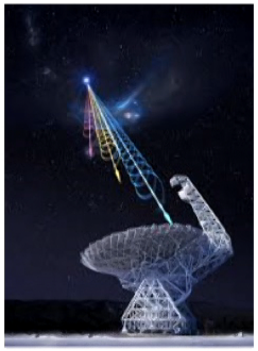
Cosmology:

- 3D "super CMB"
- most powerful way (?) to measure many cosmological parameters (early universe, neutrinos, dark matter, etc.)



Pulsars:

- new tests of GR
- new probe of gravity waves
- rich astrophysics



Fast radio bursts:

- what are they?
- potential applications…?