# Computing Betti Tables with HTCondor

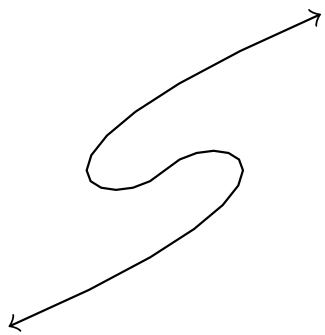
Jay Yang

May 18, 2016

# What are Betti Tables

- A tool in Algebraic Geomtry
- A shape determines a "barcode" of integers known as a Betti table.

Example



| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 6 | 8 | 3 |

- Unfortunately we don't understand the dictionary between shapes and "barcodes"
- The goal is to understand how this correspondence encodes geometry

# Role of Computation in Pure Math


## Examples

- Twin Primes Conjecture
- Riemann Hypothesis
- Average Rank of Elliptic Curves

# Our Problem

- Betti tables of the projective plane of degree d

| $d$ | Pen and Paper |
|---|---|
| 2 | Minutes |
| 3 | Hours |
| 4 | Impractical |

# Old Algorithm

## Gröbner Basis

- Based on polynomial algebra
- Developed in the 60s
- Implemented in the 80s
- Advantages
  - Already implemented
  - Well optimized
- Disadvantages
  - Difficult to distribute
  - Doubly exponential

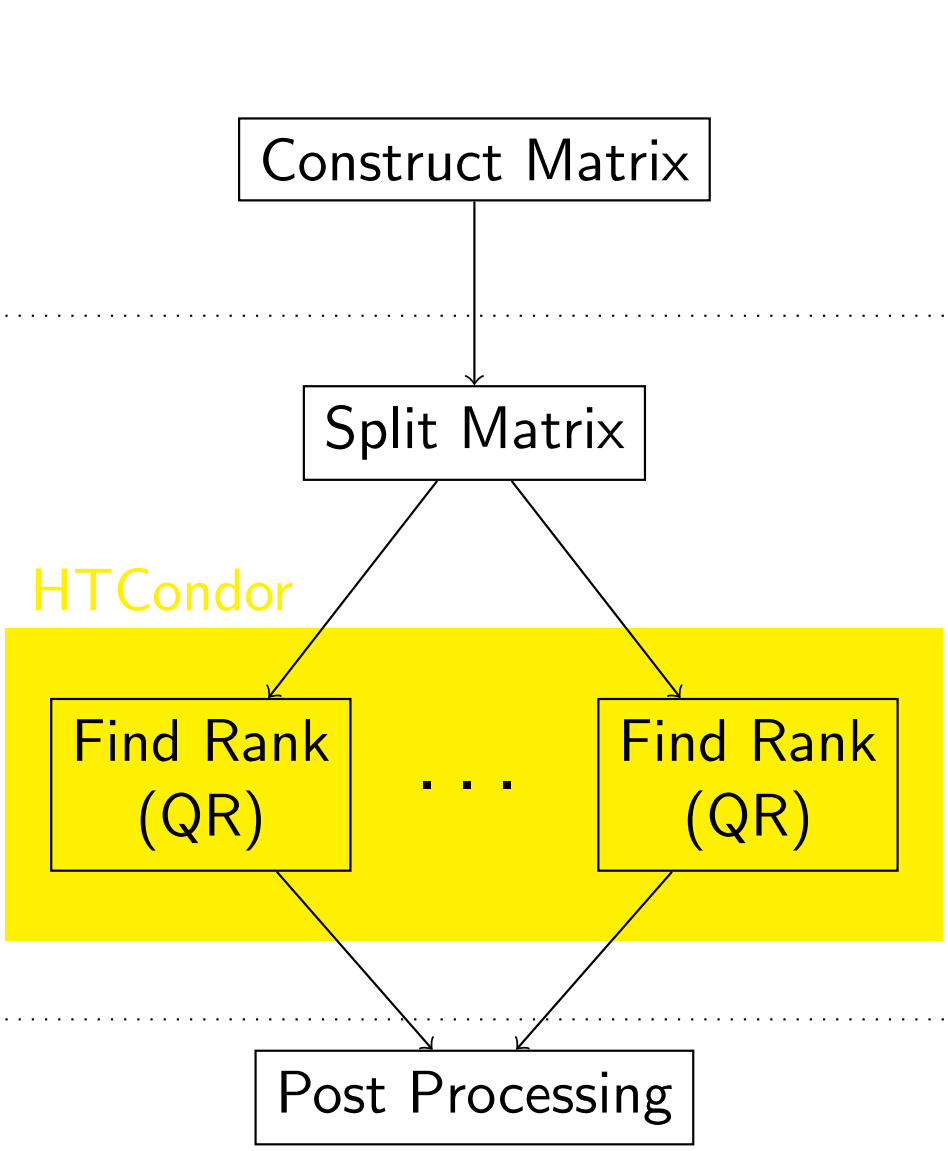
# With Gröbner Bases

| $d$ | Pen and Paper | Gröbner Basis |
|---|---|---|
| 2 | Minutes | 0.0005s |
| 3 | Hours | 0.007s |
| 4 | Impractical | 115s |
| 5 | | Out of Memory |

# New Algorithm

- Based on linear algebra
- Advantages
  - Based on well known linear algebra algorithms
  - Easily Distributable
- Disadvantages
  - Lose exactness

# Workflow




For d=5

6 matrices

600 submatrices

1 betti table (3x19 array)

# With Our Algorithm

| $d$ | Pen and Paper | Gröbner Basis | Our Algorithm |
|---|---|---|---|
| 2 | Miniutes | 0.0005s | $\sim$20s |
| 3 | Hours | 0.007s | $\sim$1m |
| 4 | Impractical | 115s | $\sim$2m |
| 5 | | Out of Memory | $\sim$11m |
| 6 | | | ?? |

# Timing Data

| $d$ | Construct Matrices | (Wall Time) | (CPU time) |
|---|---|---|---|
| 2 | 0.1s | $\sim$20s | $\sim$10s |
| 3 | 0.8s | $\sim$1m | $\sim$20s |
| 4 | 30s | $\sim$2m | $\sim$5m |
| 5 | 8m | $\sim$11m | $\sim$40m |
| 6 | 4h | ? | ? |

# Submit File

```
executable = wrapper.sh
output = outdir/single_entry_14_1.$(CLUSTER).$(PROCESS).out
error = outdir/single_entry_14_1.$(CLUSTER).$(PROCESS).err
log = single_entry_14_1.$(CLUSTER).log

universe = vanilla

arguments=$(infile) ./out_14_1/

request_memory = 6G

queue infile matching files ./matrices/map_14_1/*.dat
```

# What's Next

- Run $d = 6$
  - Largest matrices for $d = 5$ use 5GB of ram
  - Largest matrices for $d = 6$ use 10-100GB? of ram
  - Dynamic Memory Requests in Condor
  - Flock to CHTC's HTCondor pool on campus
- Obtain partial tables for $d > 6$
- Investigate other rank algorithms other than QR
- More complex shapes
- Create a database of Betti tables

# Thanks


## Thanks to

- Thanks to my collaborators Daniel Erman and David Bruce
- Steve Goldstein for introducing us to HTCondor and helping us understand and use it
- Steve Wright for his advice on matrix rank for sparse matrices
- The organizers for giving us the opportunity to give this talk