LEO SINGER, RESEARCH ASTROPHYSICIST
NASA GODDARD SPACE FLIGHT CENTER

TUTORIAL: RAPID LOCALIZATION OF COMPACT BINARY MERGERS WITH

BAYESTAR

THE INTER-UNIVERSITY CENTRE FOR ASTRONOMY AND ASTROPHYSICS (IUCAA)

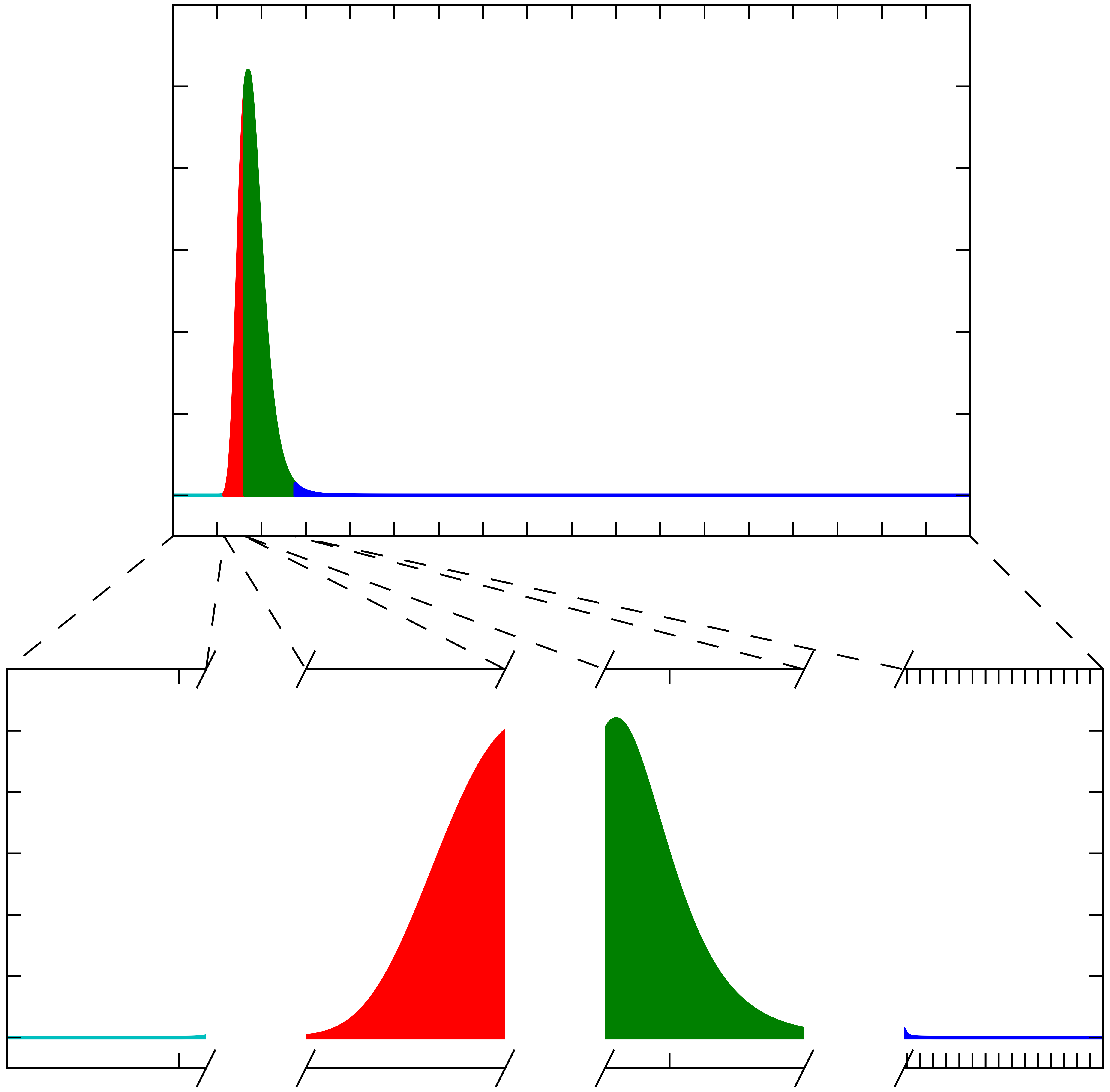
PUNE, INDIA — 24 MAY 2017

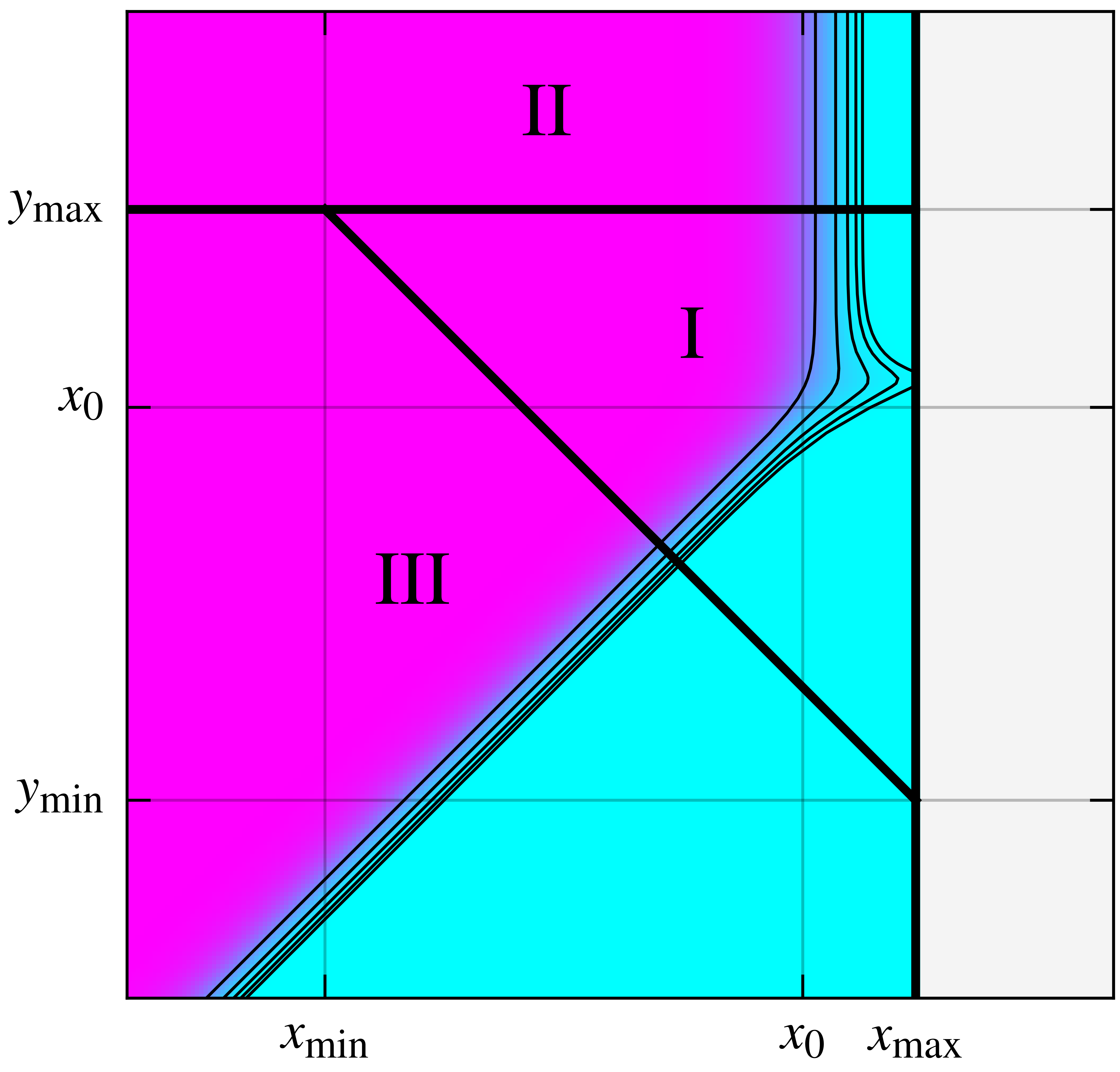
LIGO-G1700978-V2

```
*/

/*
 * Copyright (C) 2013-2017  Leo Singer
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with with program; see the file COPYING. If not, write to the
 * Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,
 * MA  02111-1307  USA
 */

#include "config.h"
#include "bayestar_cosmology.h"
#include "bayestar_sky_map.h"
#include "bayestar_distance.h"
#include "bayestar_moc.h"

#include <assert.h>
#include <float.h>
#include <math.h>
```

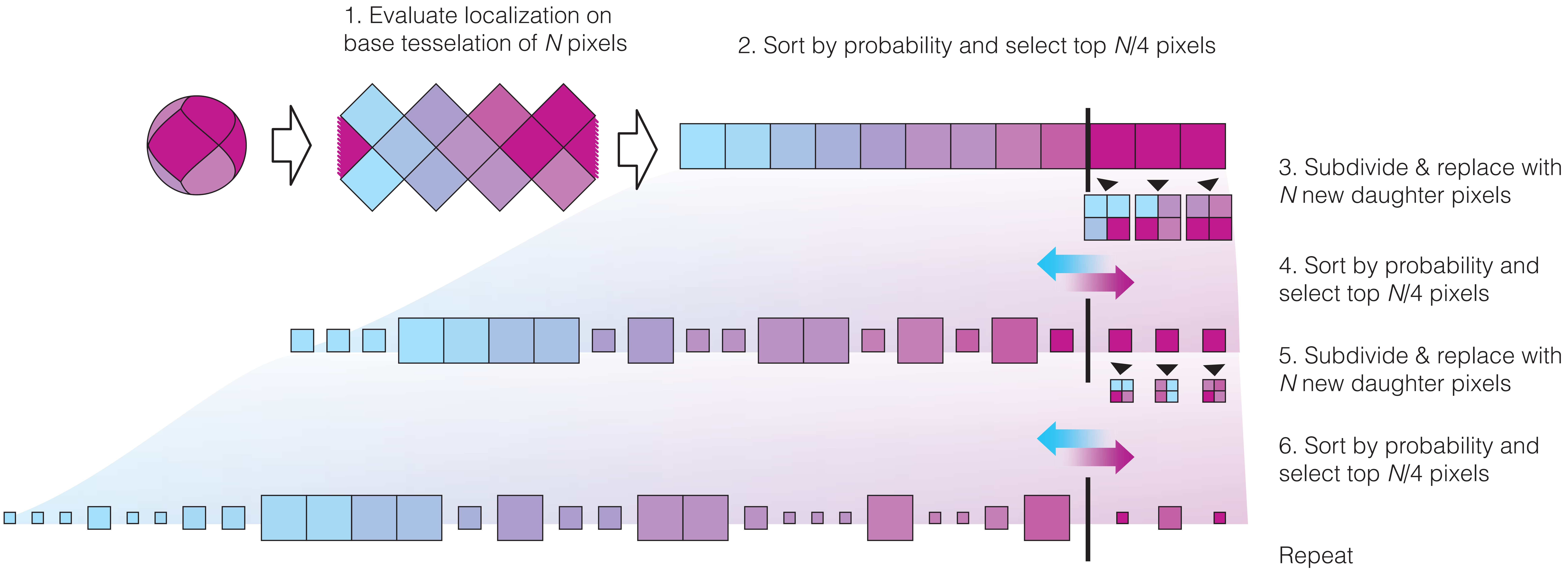
(BLACKBOARD)

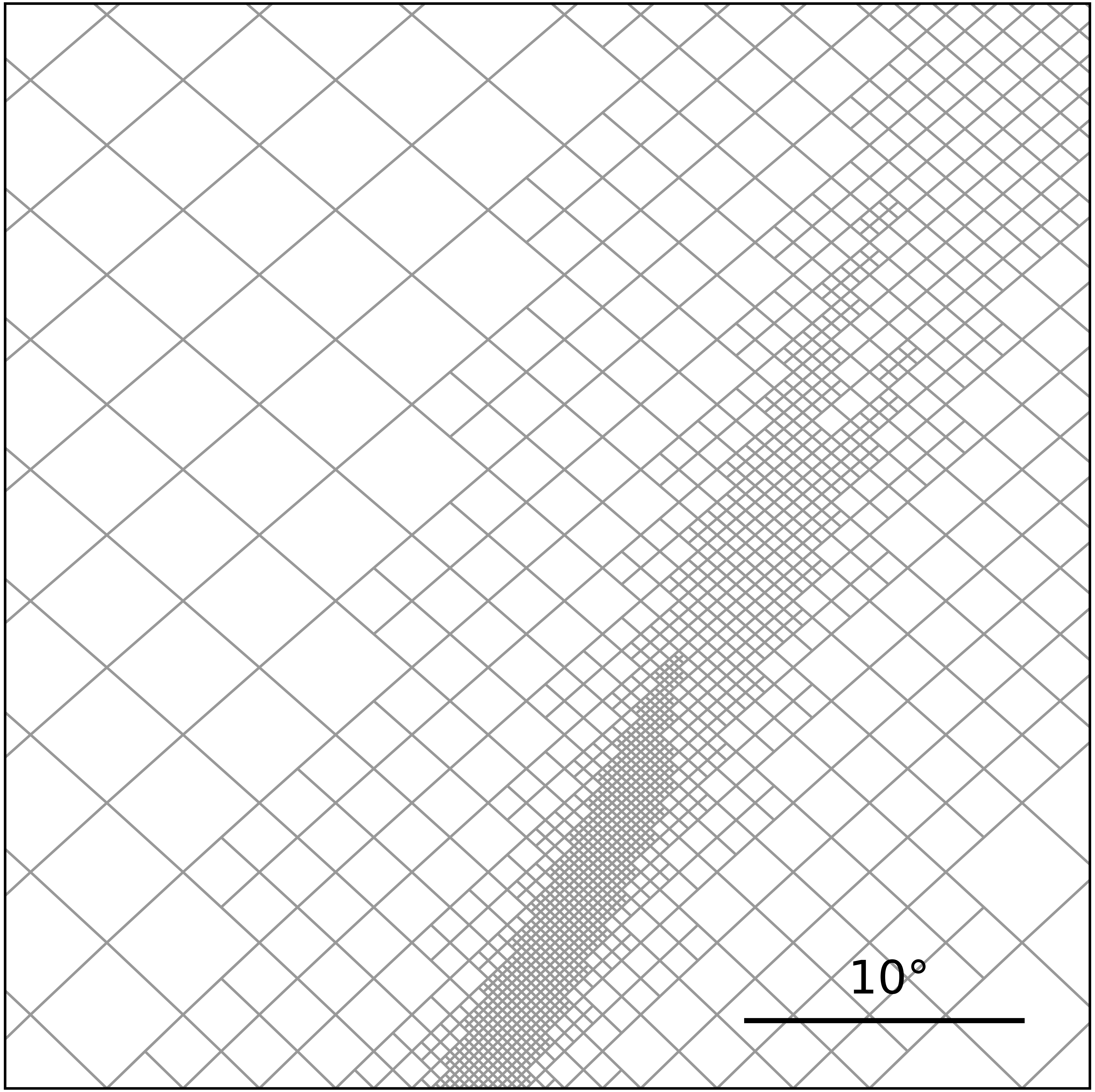
PRELIMINARIES





II
I
III
y_{max}
x_0
y_{min}
x_{min}
x_0
x_{max}


1. Evaluate localization on
base tesselation of N pixels
2. Sort by probability and select top N/4 pixels
3. Subdivide & replace with
N new daughter pixels
4. Sort by probability and
select top N/4 pixels
5. Subdivide & replace with
N new daughter pixels
6. Sort by probability and
select top N/4 pixels
Repeat


10°

HOW TO INSTALL BAYESTAR

RECOMMENDED: LIGO DATA GRID CLUSTERS

- BAYESTAR is part of LALSuite. The latest stable release of LALSuite is installed on all LIGO Data Grid (LDG) clusters. This version is used for the production, on-line analysis.
- To use, simply log in to any LDG cluster head node. Examples:

```
ldas-pcdev1.ligo.caltech.edu
pcdev1.phys.uwm.edu
atlas1.atlas.aei.uni-hannover.de
...
```

- Caveats:
  - The LALSuite installation on the IUCAA cluster is not working today.
  - A common pitfall is that if you have set the `PATH`, `PYTHONPATH`, `LD_LIBRARY_PATH`, etc. in your `.profile` script, these may override the system LALSuite installation.

RECOMMENDED FOR MACS: MACPORTS

- The latest stable release of LALSuite is in MacPorts.
- If you have a Mac, but you don't have MacPorts already, then you should get it! It will save you a lot of time in setting up all of the dependencies that you need for LALSuite software development on your own machine.
- Get MacPorts from https://macports.org/.
- Install LALInference, which includes BAYESTAR, using the following command:

```
sudo port install lalapps pylal py27-lalinference
```

- Go get a cup of tea/coffee while it builds.

OTHER OPTIONS FOR DEBIAN AND SCIENTIFIC LINUX

- If you have a Debian or Scientific Linux computer, then you can add the LSCSoft repositories to your package manager and install the same builds that are used on the LDG clusters.
- See https://wiki.ligo.org/DASWG/SoftwareDownloads for instructions.

ADVANCED: INSTALL LALSUITE FROM SOURCE

- For advanced users. This is the best option ***only*** if you plan to contribute to LALSuite software development.
- Download source releases from: http://software.ligo.org/lscsoft/source/
- Or check out bleeding-edge code from git:
  git clone git://versions.ligo.org/lalsuite.git
- Highly recommended shortcut: use JHbuild for build and environment automation. See http://lpsinger.github.io/modulesets/

ONLINE ANALYSIS

HOW TO RUN BAYESTAR

Run BAYESTAR on GraceDB events

NOTE: for this to work, you must first authenticate your LIGO.ORG account by running `ligo-proxy-init`.

IMPORTANT! Use the `--dry-run` option to save sky maps to disk instead of uploading the to GraceDB.

Run on this GraceDB ID. This is GW151226.

bayestar_localize_lvalert

```
bayestar_localize_lvalert \

--dry-run \

G211117
```

output: bayestar.fits.gz

A SMALL SIMULATION STUDY

HOW TO RUN BAYESTAR

Create a pseudorandom sample of injections

Write a table of CBC injections to `inj.xml`.

Mass distribution: in this example, pinned to 1.4 and 1.4 solar masses.

Coalescence time distribution: adjust time step, start, and stop time control the number of injections.

Distance distribution: uniform in Euclidean volume. WARNING: distances are in kpc!

Sky position and inclination distribution.

Waveform options: typically use `TaylorF2threePointFivePN` for BNS.

lalapps_inspinj

```
lalapps_inspinj -o inj.xml \

--m-distr fixMasses \
--fixed-mass1 1.4 --fixed-mass2 1.4 \

--t-distr uniform --time-step 7200 \
--gps-start-time 1000000000 \
--gps-end-time 1000086400 \

--d-distr volume \
--min-distance 1 --max-distance 600e3 \

--l-distr random --i-distr uniform \

--f-lower 30 --disable-spin \
--waveform TaylorF2threePointFivePN
```

output: inj.xml

Synthesize noise PSDs from models in LALSimulation

Write discretely sampled PSDs to `psd.xml`.

Specify noise models for desired detectors.

Optional: apply scale factor to selected PSDs to increase or decrease their sensitivity. The horizon distance goes as one over scale squared.

bayestar_sample_model_psd

```
bayestar_sample_model_psd -o psd.xml \

--H1=aLIGOZeroDetHighPower \
--L1=aLIGOZeroDetHighPower \
--I1=aLIGOZeroDetHighPower \
--V1=AdvVirgo \
--K1=KAGRA \

--I1-scale=0.75
```

output: psd.xml

Simulated detection pipeline: transform injections into coincidences

bayestar_realize_coincs

Write output to `coinc.xml`.

```
bayestar_realize_coincs -o coinc.xml \
```

Use the injections and noise PSDs that we generated.

```
inj.xml --reference-psd psd.xml \
```

Specify which detectors are in science mode.

```
--detector H1 L1 V1 I1 K1 \
```

Optionally, adjust the detection threshold: single-detector SNR, network SNR, and minimum number of detectors above threshold to form a coincidence.

```
--snr-threshold 4.0 \
--net-snr-threshold 12.0 \
--min-triggers 2 \
```

Optionally, save triggers that were below the single-detector threshold.

```
--keep-subthreshold
```

output: coinc.xml

Match injections and coincidences, save as SQLite database

ligolw_add, ligolw_inspinjfind, ligolw_sqlite

Concatenate `coinc.xml` and `inj.xml` into one XML file.

```
ligolw_add coinc.xml inj.xml -o coinc_inj.xml
```

Injection finding: match coincidences to injections.

```
ligolw_inspinjfind \
< coinc_inj.xml > coinc_inj_found.xml
```

Convert XML to SQLite database. WARNING: do not forget the `--preserve-ids` and `--replace` options!

```
ligolw_sqlite --preserve-ids --replace --database
coinc_inj_found.sqlite coinc_inj_found.xml
```

Or do all three steps at once by piping the commands together.

```
ligolw_add coinc.xml inj.xml \
| ligolw_inspinjfind \
| ligolw_sqlite --preserve-ids --replace --database \
coinc_inj_found.sqlite /dev/stdin
```

output: coinc_inj_found.sqlite

Run BAYESTAR sky localization on a batch of coincidences

IMPORTANT: HIGHLY RECOMMENDED IF USING A SHARED WORKSTATION. Explicitly set the number of OpenMP threads instead of using all available cores.

Run BAYESTAR on all coincident events in `coinc.xml`.

NEW on master! Submit jobs to Condor instead of running BAYESTAR locally.

Output FITS files are saved with names that correspond to the numeric part of the `coinc_event_id` column in the `coinc.xml` file.

bayestar_localize_coincs

```
export OMP_NUM_THREADS=4
```

```
bayestar_localize_coincs coinc.xml \
--condor-submit
```

output: 0.toa_phoa_snr.fits, 1.toa_phoa_snr.fits, ...

Analyze a batch of injections and sky maps

Save output as an ASCII table in the file `bayestar.out`.

Read events from the SQLite database that we created; read all sky maps in this directory.

Optional: calculate the 50% and 90% credible areas.

Optional: calculate the probability contained within the smallest credible regions of 10 and 100 deg^2.

Optional: count the number of disjoint patches on the sky. WARNING: this option makes the script very slow!

Optional, but highly recommended: analyze sky maps using multiple threads. In this example, we use 8 worker processes.

bayestar_aggregate_found_injections

```
bayestar_aggregate_found_injections -o bayestar.out \
coinc_inj_found.sqlite *.fits.gz \
--contour 50 90 \
--area 10 100 \
--modes \
-j 8
```

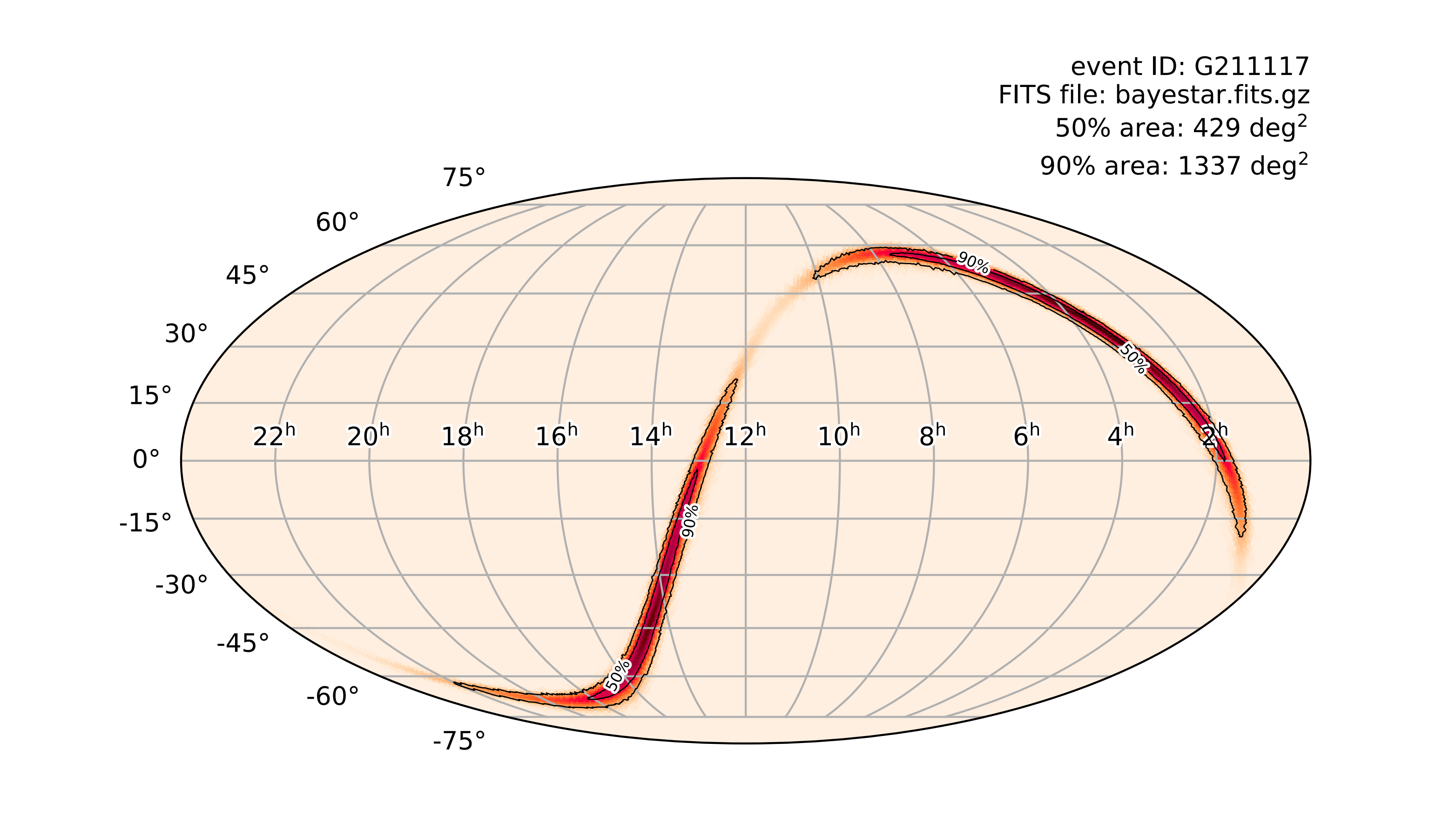
output: bayestar.out

Full example

Makefile

```
all: bayestar.out

inj.xml:
	lalapps_inspinj -o inj.xml \
	--m-distr fixMasses --fixed-mass1 1.4 --fixed-mass2 1.4 \
	--t-distr uniform --time-step 7200 \
	--gps-start-time 1000000000 --gps-end-time 1000086400 \
	--d-distr volume --min-distance 1 --max-distance 600e3 \
	--l-distr random --i-distr uniform \
	--f-lower 30 --waveform TaylorF2threePointFivePN --disable-spin

psd.xml:
	bayestar_sample_model_psd -o psd.xml \
	--H1=aLIGOZeroDetHighPower \
	--L1=aLIGOZeroDetHighPower \
	--I1=aLIGOZeroDetHighPower \
	--V1=AdvVirgo \
	--K1=KAGRA

coinc.xml: psd.xml inj.xml
	bayestar_realize_coincs --reference-psd psd.xml inj.xml -o coinc.xml \
	--detector H1 L1 V1 I1 K1 --keep-subthreshold

coinc_inj_found.sqlite: coinc.xml inj.xml
	ligolw_add coinc.xml inj.xml | \
	ligolw_inspinjfind | \
	ligolw_sqlite  --preserve-ids --replace \
	--database coinc_inj_found.sqlite /dev/stdin

0.toa_phoa_snr.fits.gz: coinc.xml psd.xml
	bayestar_localize_coincs coinc.xml

bayestar.out: coinc_inj_found.sqlite 0.toa_phoa_snr.fits
	bayestar_aggregate_found_injections \
	coinc_inj_found.sqlite \*.fits.gz -o bayestar.out \
	--contour 50 90

clean:
	rm -f inj.xml psd.xml coinc.xml coinc_inj_found.sqlite *.fits bayestar.out
```

SOME OTHER USEFUL TOOLS



Plot a sky map

Plot the sky map `bayestar.fits.gz`.

Optionally save the image to `bayestar.pdf`. If omitted, then display the image in a window (requires desktop environment, X11, or equivalent).

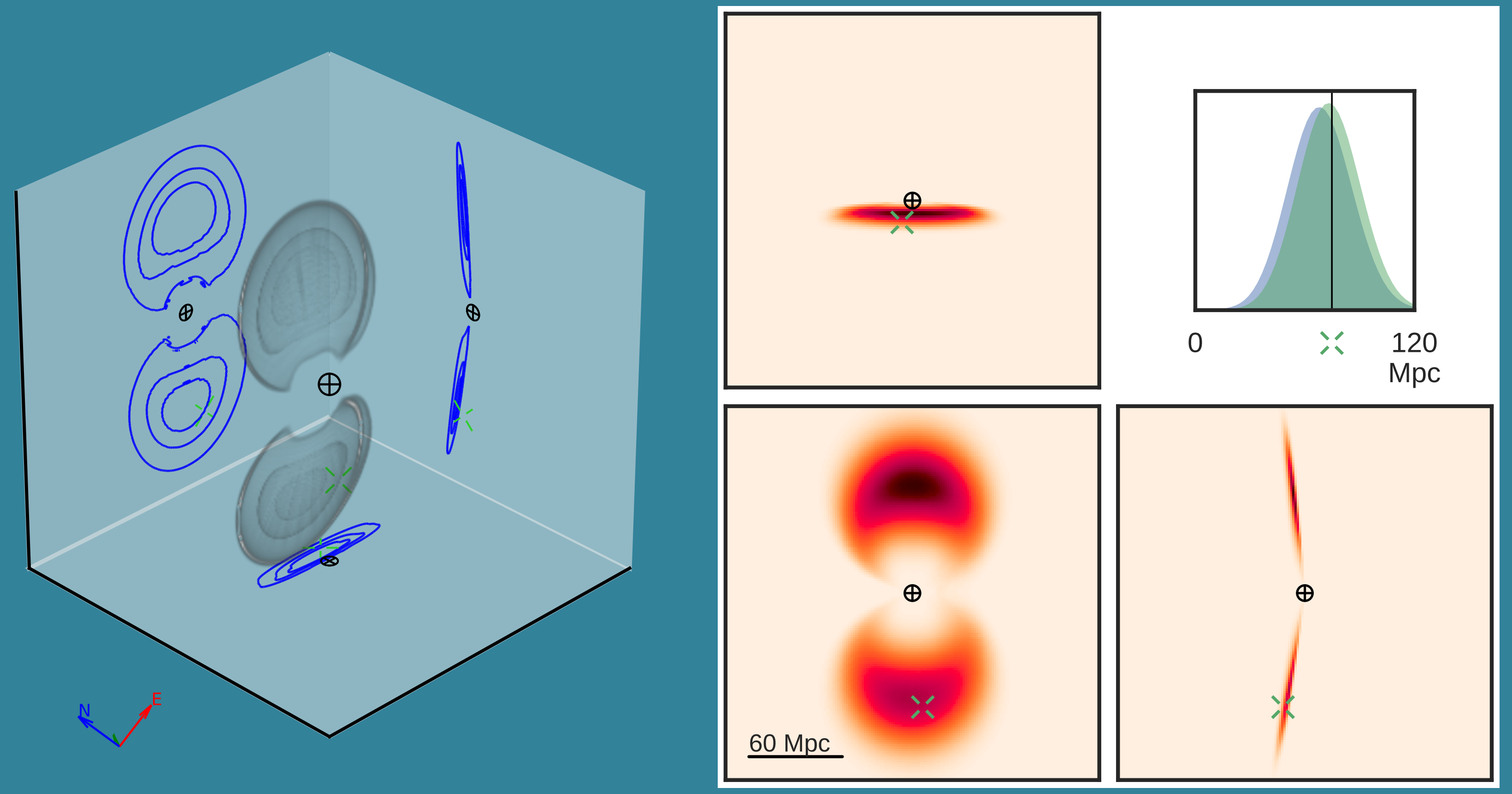
Optionally label the plot with the FITS filename and the contour areas.

Optionally plot the 50% and 90% contours.

bayestar_plot_allsky

```
bayestar_plot_allsky bayestar.fits.gz \

-o bayestar.pdf \

--annotate \

--contour 50 90
```

output: bayestar.pdf



Plot a sky map in 3D using volume rendering

IMPORTANT: HIGHLY RECOMMENDED IF USING A SHARED WORKSTATION. Explicitly set the number of OpenMP threads instead of using all available cores.

Plot the sky map `bayestar.fits.gz`.

Optionally save the image to `bayestar3d.pdf`. If omitted, then display the image in a window (requires desktop environment, X11, or equivalent).

bayestar_plot_volume

```
export OMP_NUM_THREADS=4
```

```
bayestar_plot_volume bayestar.fits.gz \
-o bayestar3d.pdf
```

output: bayestar3d.pdf

REFERENCES

- Singer, L. P., Price, L R., & Farr, B., et al., 2014, The First Two Years of Electromagnetic Follow-up with Advanced LIGO and Virgo. *Astrophysical Journal*, 795, 2, 105. https://arxiv.org/abs/1404.5623, https://dx.doi.org/10.1088/0004-637X/795/2/105

- Singer, L. P. & Price, L. R., 2016, Rapid Bayesian Position Reconstruction for Gravitational-Wave Transients. *Physical Review D*, 93, 024013. https://arxiv.org/abs/1508.03634, https://dx.doi.org/10.1103/PhysRevD.93.024013

- Singer, L. P., Chen, H.-Y., Holz, D. E., et al., 2016, Going the Distance: Mapping Host Galaxies of LIGO and Virgo Sources in Three Dimensions Using Local Cosmography and Targeted Follow-up. *Astrophysical Journal Letters*, 829, L15. https://arxiv.org/abs/1603.07333, https://dx.doi.org/10.3847/2041-8205/829/1/L15

- Singer, L. P., Chen, H.-Y., Holz, D. E., et al., 2016, Supplement: Going the Distance: Mapping Host Galaxies of LIGO and Virgo Sources in Three Dimensions Using Local Cosmography and Targeted Follow-up. *Astrophysical Journal Supplement Series*, 226, 1, 10. https://arxiv.org/abs/1605.04242, https://dx.doi.org/10.3847/0067-0049/226/1/10

- http://ligo.org/scientists/first2years/

- http://asd.gsfc.nasa.gov/Leo.Singer/going-the-distance/