StreamSim tutorial

Matthieu Pélissier (2nd year PhD student)
Supervised by Marine Kuna and David Maurin
With Alex Drlica Wagner and Peter Ferguson

DESC project

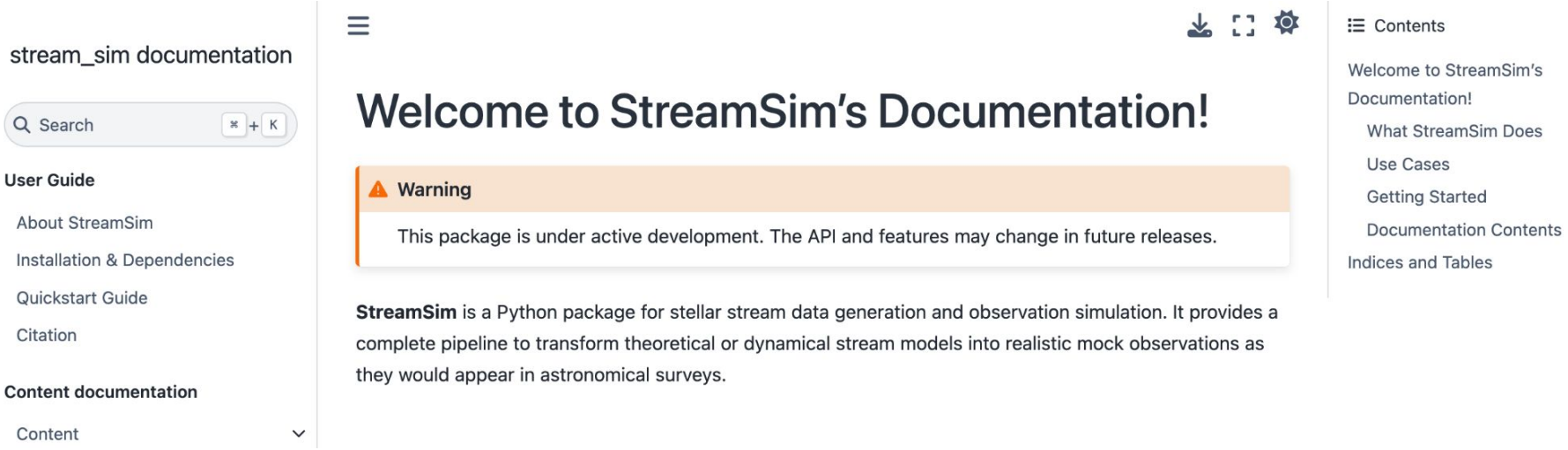
Table of content

Introduction

Introduction: what is StreamSim?

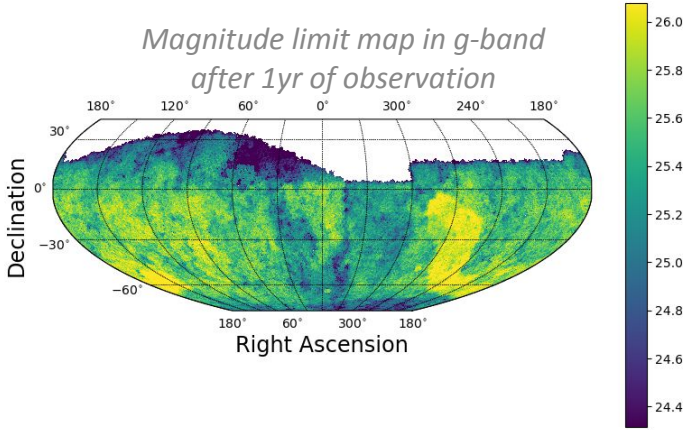
StreamSim:

- **Public** package available on GitHub
- Currently under active **development**
- Designed for **easy** use (hopefully!)
- **Documentation** and **tutorials** available



Introduction: why do we need StreamSim?

Dynamical simulations = idealized situation



- Observation **impacted** by survey's **systematics**
- **Not all** the star members are **observed**

StreamSim goals:

- **Bridge** dynamical **simulation** and **observation** in photometric surveys
- **Not** intended to generate **dynamic models** or N-body simulations

Dynamical simulations → StreamSim → Observed data

Mimic LSST observation (completeness, magnitudes and errors according to depth maps)

StreamSim functionalities

Generating mocks: parametric sampler

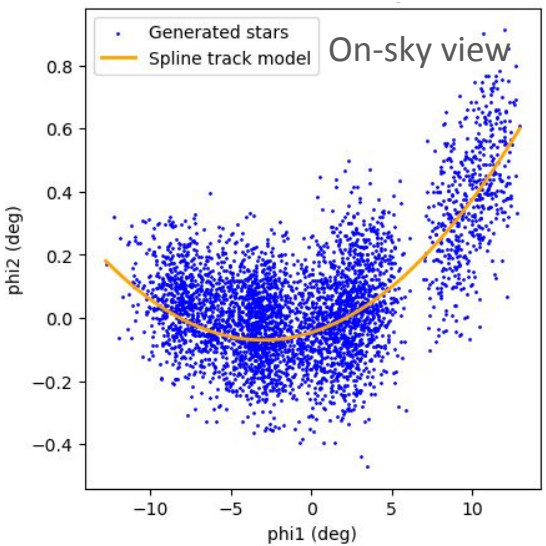
Parametric models (e.g splines) defining:

- Track
- Density profile
- Distance modulus

Sample N_stars

Or your **dynamical simulations**

Stars coordinates (phi1, phi2, distance modulus)



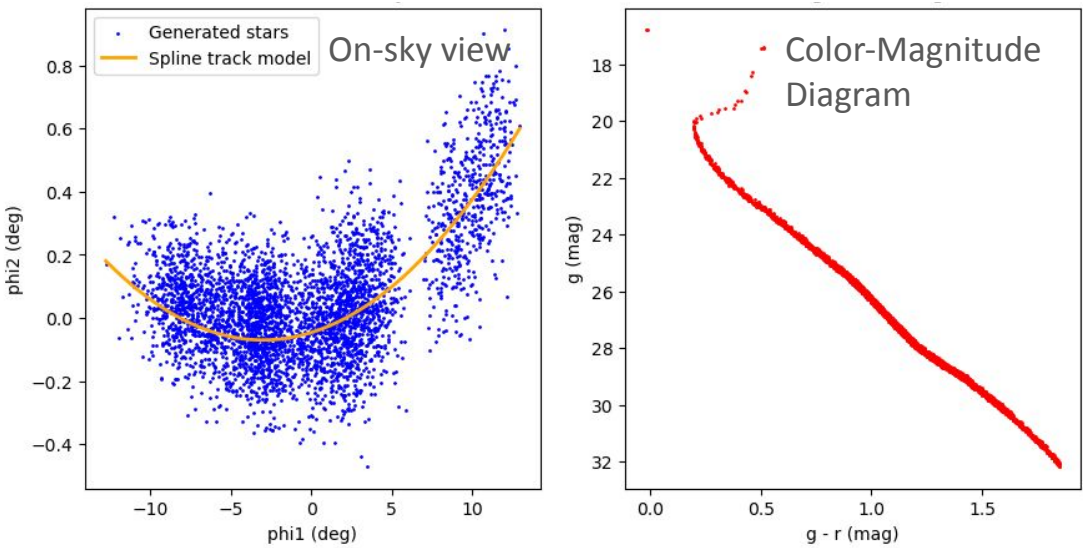
Example: realization from Phoenix stream spline

Generating mocks: magnitudes

Parametric models (e.g splines) defining:

- Track
- Density profile
- Distance modulus

Sample N_stars



Example: realization from Phoenix stream spline

Or your **dynamical simulations**

Stars coordinates (phi1, phi2, distance modulus)

Choose stars population **Age + metallicity**

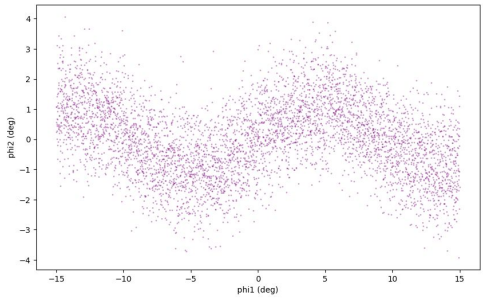
Define Isochrone (using Ugali package)

= **Distribution of a star population** in a Colour-Magnitude Diagram

Sample true apparent magnitudes in chosen **photometric bands**

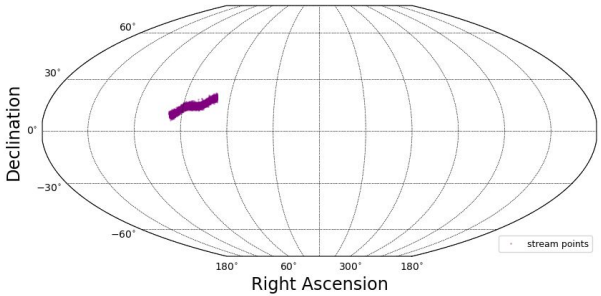
Generating mocks: coordinates conversion

Stream generated in analysis coordinates **(phi1, phi2)**



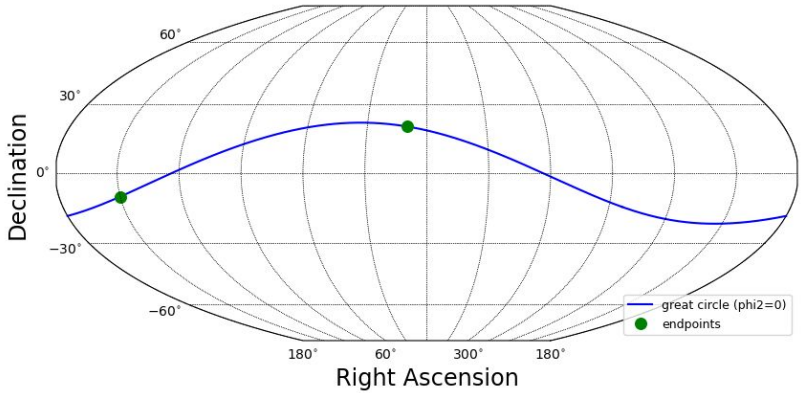
Conversion = rotation of basis

Stream observed in **(ra,dec)**



- choose 2 **endpoints** (anywhere in the sky or in a selected mask)
- They define a **great circle** (phi2 = 0)

Note: optional if (ra, dec) already in your data



Conversion to observed data-like: pipeline

(ra, dec), apparent magnitudes

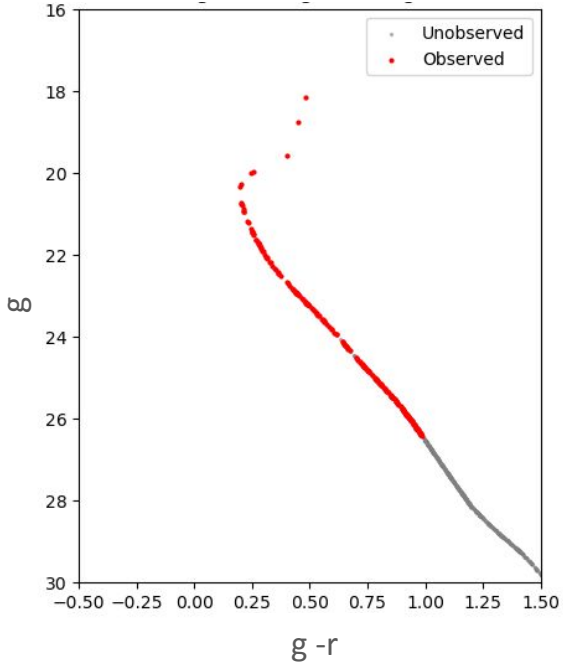
↓

Injection pipeline

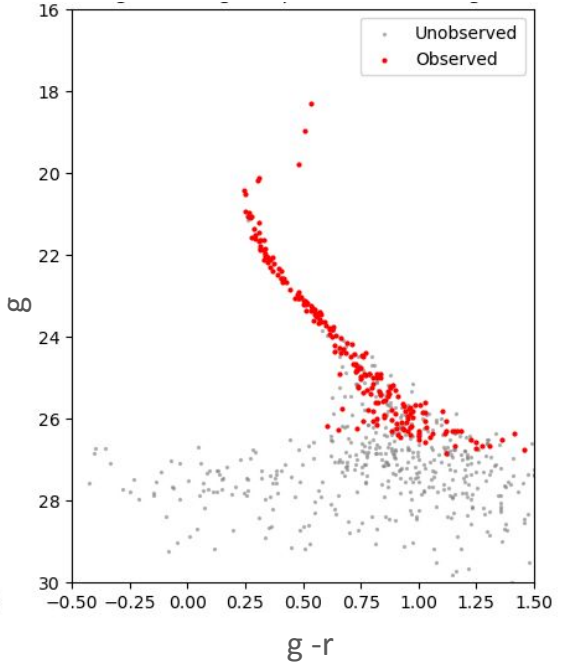
↓

Observed magnitudes + errors, detection + classification boolean

CMD with **true** magnitudes



CMD with **observed** magnitudes



Conversion to observed data-like: pipeline

(ra, dec), apparent magnitudes

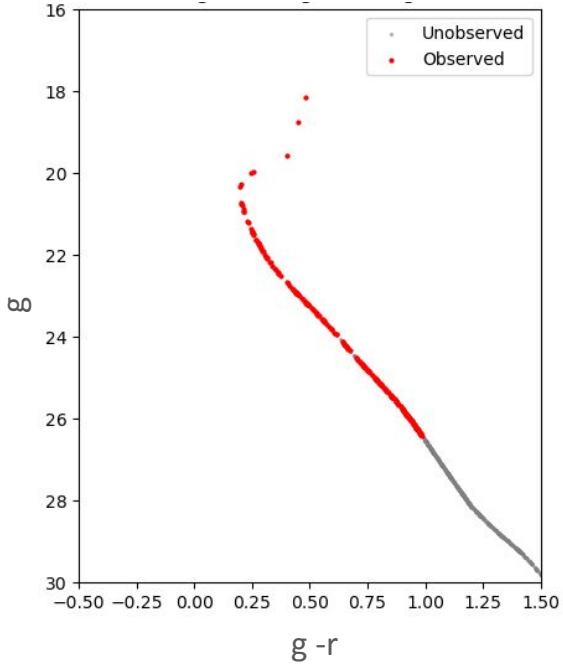
↓

Injection pipeline

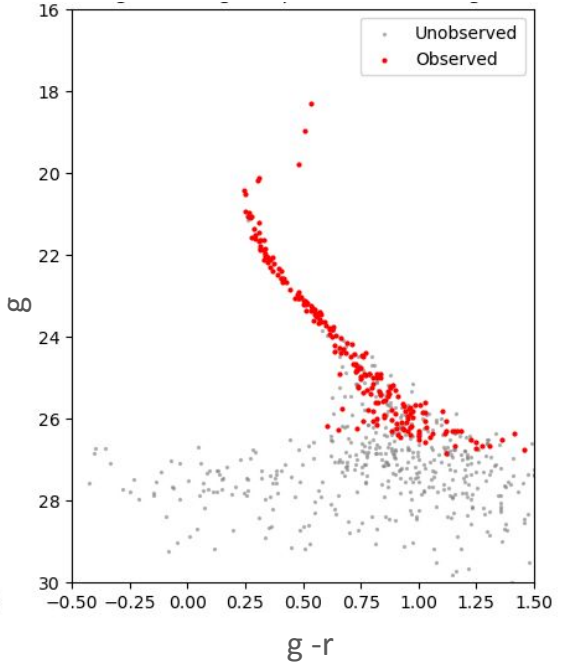
↓

Observed magnitudes + errors, detection + classification boolean

CMD with **true** magnitudes

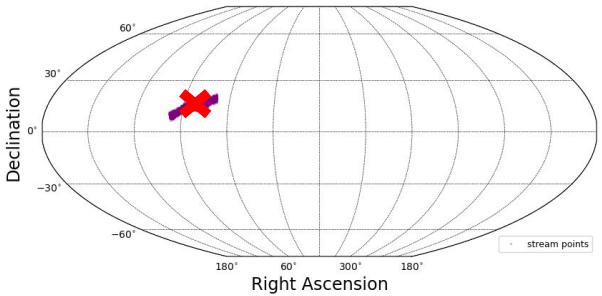


CMD with **observed** magnitudes

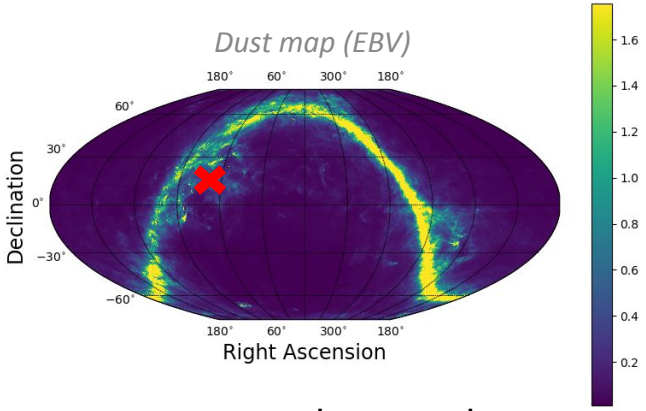


Conversion to observed data-like: dust extinction

For each stars of the stream
(ra, dec, magnitude)



Estimate **dust** extinction at (ra, dec)

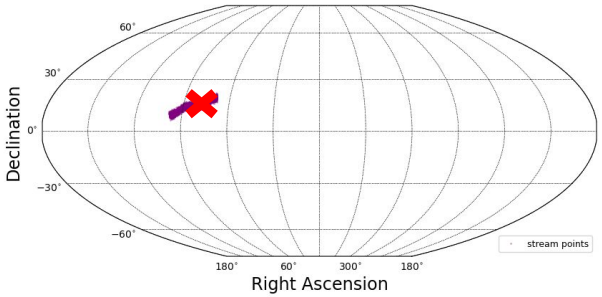


mag_apparent = mag_true + ebv × coeff

Apparent magnitude

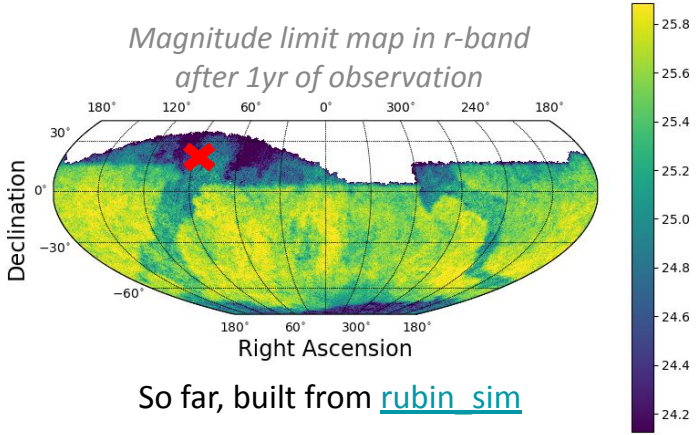
Conversion to observed data-like: magnitude limits

For each stars of the stream
(ra, dec, magnitude)



Estimate **magnitude limit** at (ra, dec)
in every bands

Magnitude limit map in r-band
after 1yr of observation

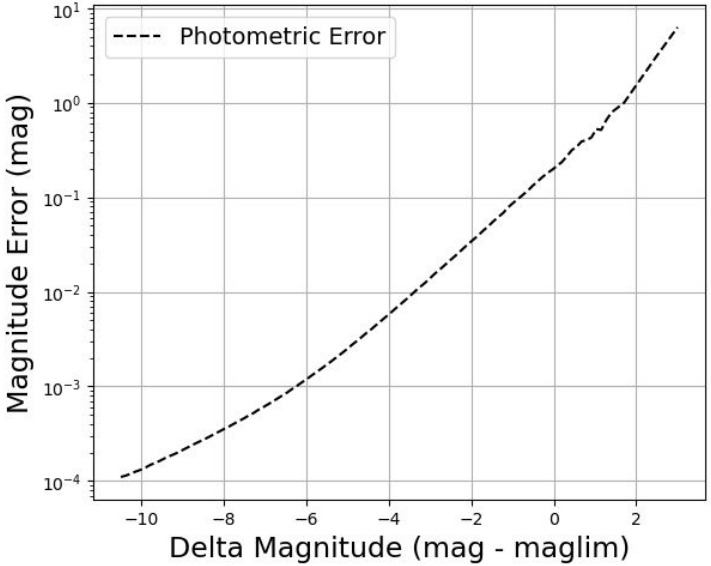


So far, built from [rubin_sim]

Delta magnitude = apparent magnitude - magnitude limit

Conversion to observed data-like: magnitude and errors

Delta magnitude → **Assign photometric errors** → **Sample observed magnitudes** (gaussian in flux)

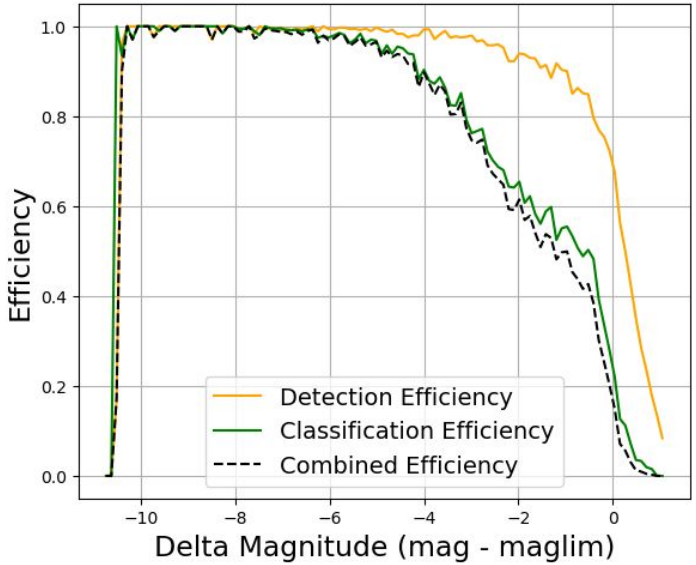


Estimated using DC2 catalog in r band (Kabelo Tsiane et al.)

Conversion to observed data-like: detection and classification

Delta magnitude

Estimated completeness



Estimated using DC2 catalog in r band (Kabelo Tsiane et al.)

Star observed if

$$Unif(0,1) < Completeness(\text{Apparent mag r})$$

\+ cut on SNR in g band

Conversion to observed data-like: summarize

Quantities estimated by StreamSim:

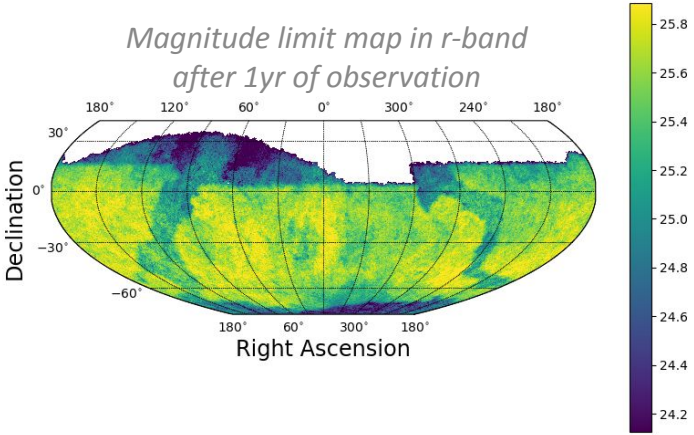
- **Coordinates** (phi1, phi2, ra, dec)
- Distance modulus
- Apparent **magnitudes**

- **Observed magnitudes**
- Photometric **errors**
- **Detection + classification** flag

Conversion to observed data-like: limits

Approximations:

- **Not** a true **synthetic source injection**
- Survey **systematics compressed** into magnitude-limit maps
- Assumes selection functions and errors shift with the magnitude limit



Advantages:

- **Low** computational **cost** → easily applied to batch of simulations
- Mimics survey selection and properties

How to use StreamSim

How to use StreamSim: resources

Documentation: available here
Including API, installation, quick start

Tutorials: available here
Notebook for the generation and injection parts

Discussions

Discussions

Future possible development:

- Add **others surveys** and **releases**: DES, DP2…
- Implement isochrone sampling in **more** than **2 bands** g and r
- Implement **velocities** sampling and the corresponding survey degradation
- Others issues (cf GitHub)

Discussions

Future possible development:

- Add **others surveys** and **releases**: DES, DP2…
- Implement isochrone sampling in **more** than **2 bands** g and r
- Implement **velocities** sampling and the corresponding survey degradation
- Others issues (cf GitHub)

Thanks for your attention

Questions? Comments?

Any specific requirements you would like to see implemented in StreamSim?

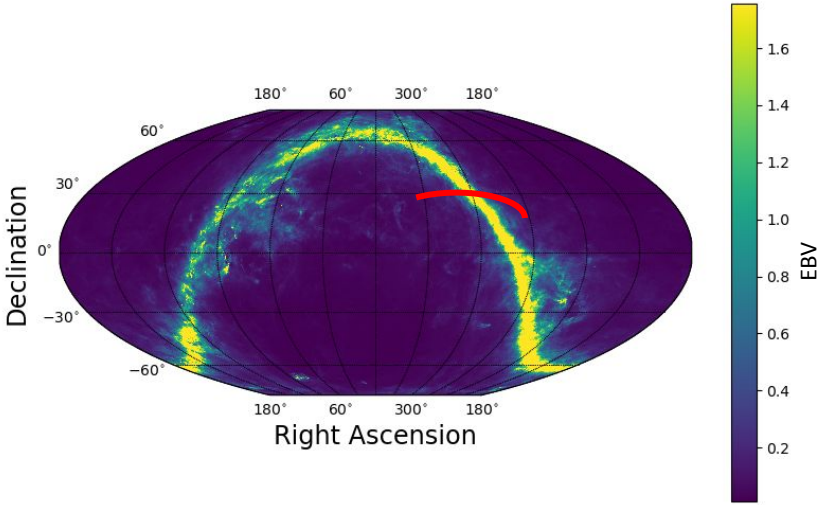
Appendix

Location injection

So far, stream were injected randomly inside the footprint → can be **crossed by dust** →

Inconvenient:

- Our pipeline will measure fluctuations that are **not due to LSST systematics** but to astrophysics
- **Not realistic**: people won't study density fluctuations of a stream crossing the galactic plane



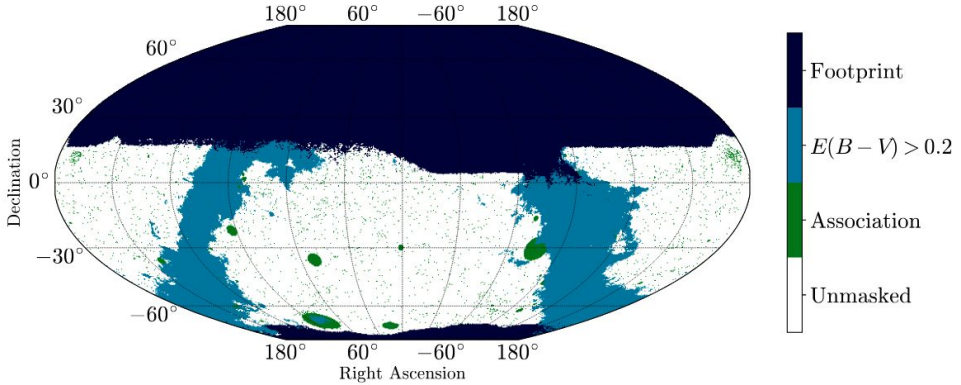


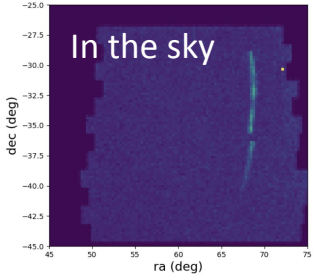
Fig 7. Kabelo Tsiane's paper

restrict the injection to the area outside dust?

IV. Annexes

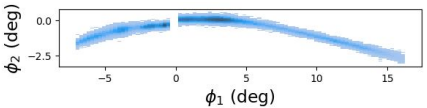
Stream rotation

- **Conversion** to preferred coordinates (ϕ1,ϕ2) to **facilitate analysis** in (ra,dec) using the Gala package



Rotation

In the analysis



(**ra, dec**) coordinates

Select **two points** in (ra,dec) that belong to the stream

It defines the **great circle** (ϕ2=0)

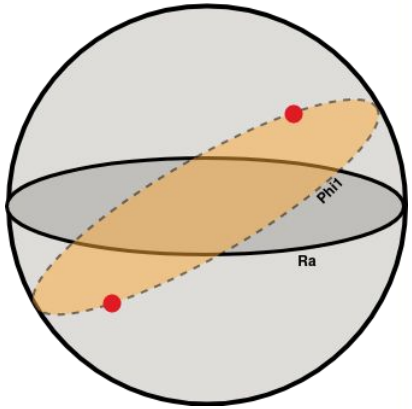
Rotation between spherical frames to **(ϕ1,ϕ2)**

(ϕ1,ϕ2) coordinates

Select **two points** in (ra,dec) that would belong to the stream (currently done **randomly** in the footprint)

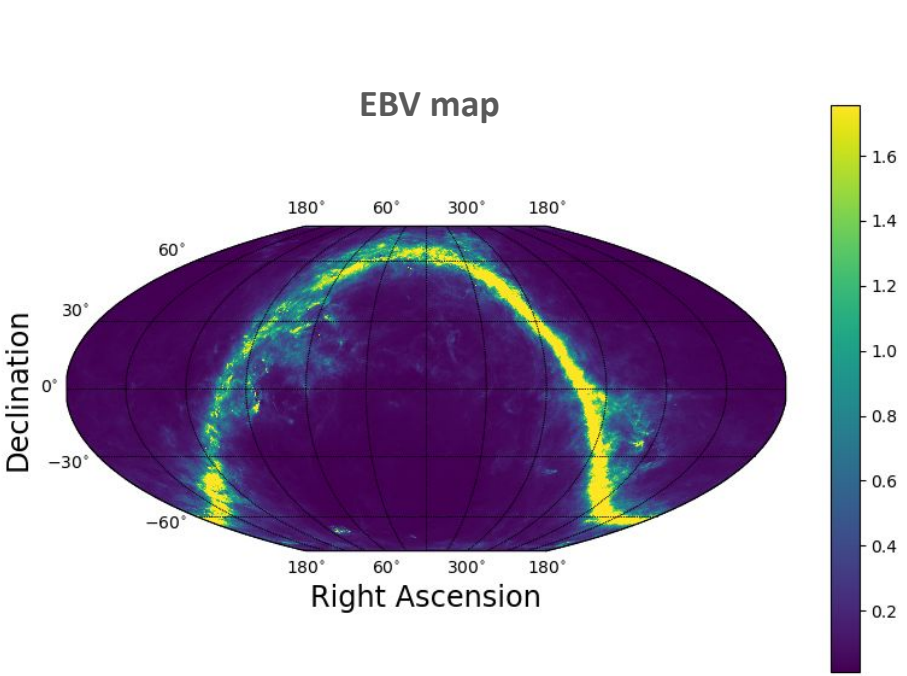
It defines the **great circle** (ϕ2=0)

Rotation between spherical frames to (**ra,dec**)



Appendix: Dust extinction estimation

Use EBV maps to obtain magnitude extinction in g and r band



x 2.273

x 3.237