Using Machine Learning Algorithm to search for gravitational waves progenitors

PhD student: Julien Marchioro (APC)
Supervisors : S. Chaty & E. Chassande-Mottin (APC)

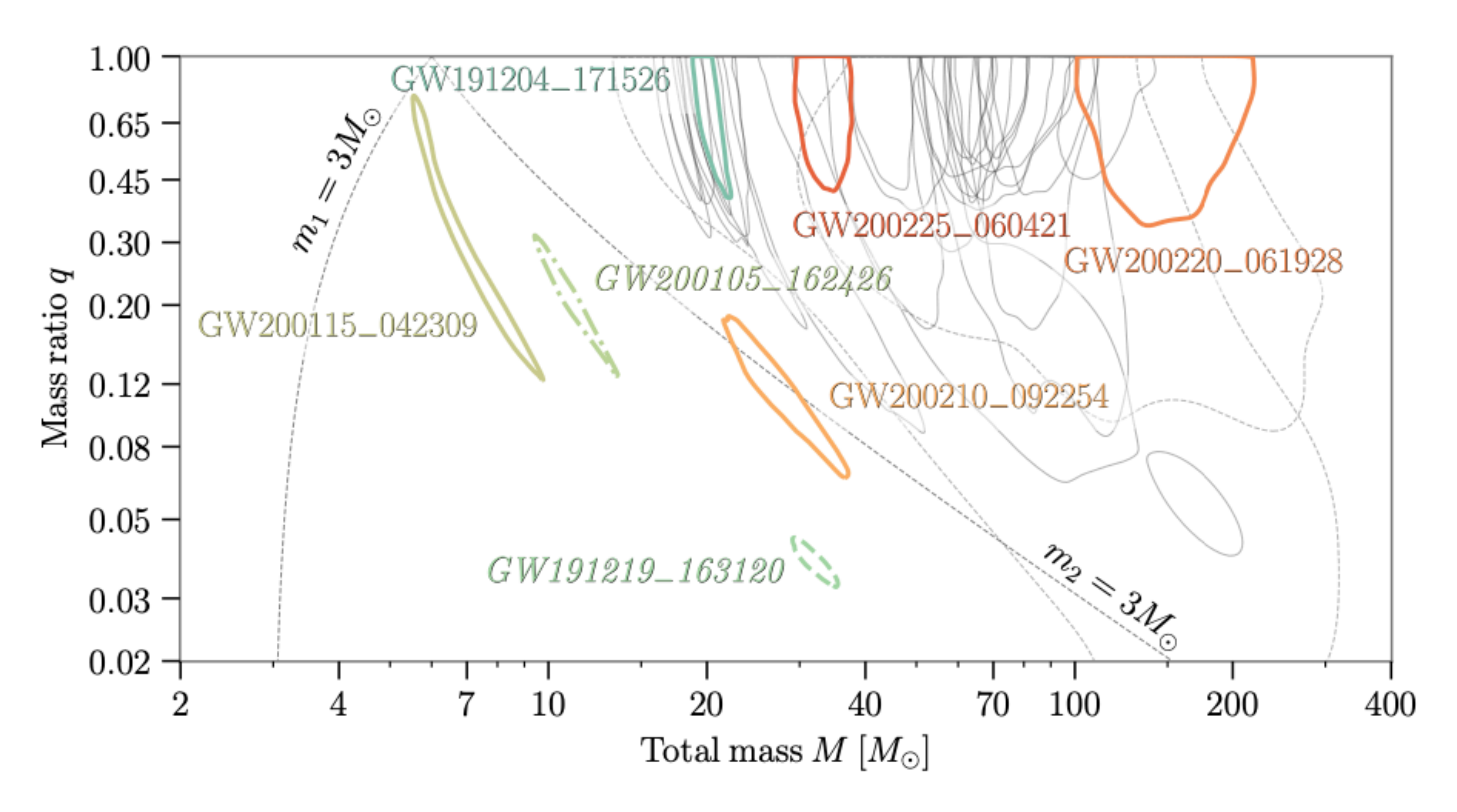
How can we train a machine learning algorithm to infer the origin of the low mass binary black holes detected in LIGO/Virgo ?

LIGO/Virgo mission

- 90 GW signals from compact binary coalescences, primarily mergers of binary black holes (BBH) — Last results from the O3 science run

- What is the origin of those BBH? Different scenarios: dynamical formation in globular clusters or isolated binaries? Are LIGO/Virgo observations (BH mass or spins) compatible with those scenarios ?

- Idea of this talk : Focus un isolated binaries and make prediction about their binary star progenitors

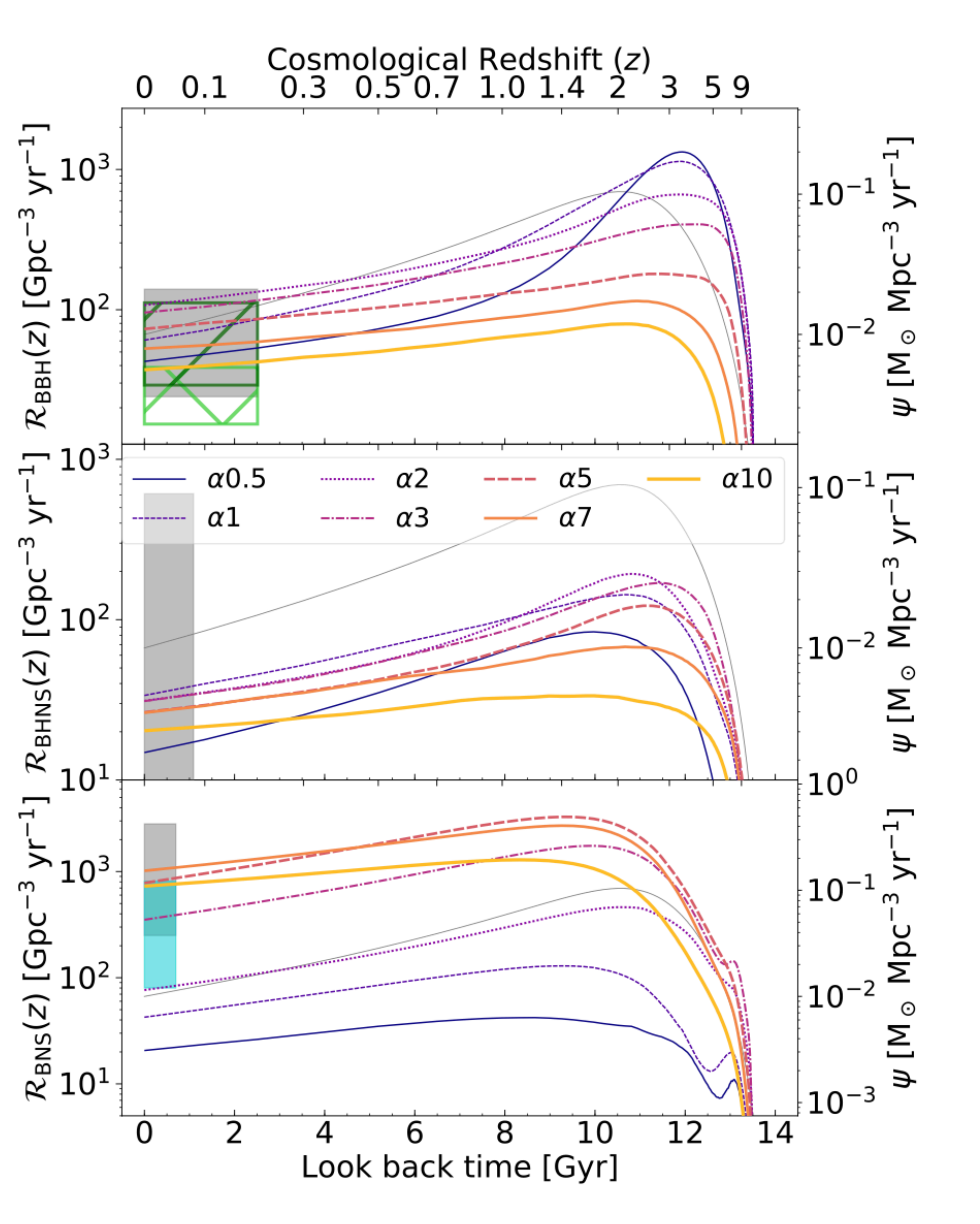


Total mass vs Mass ratio of the GWTC-3 detected events. Credits Abbott et al. (2021)

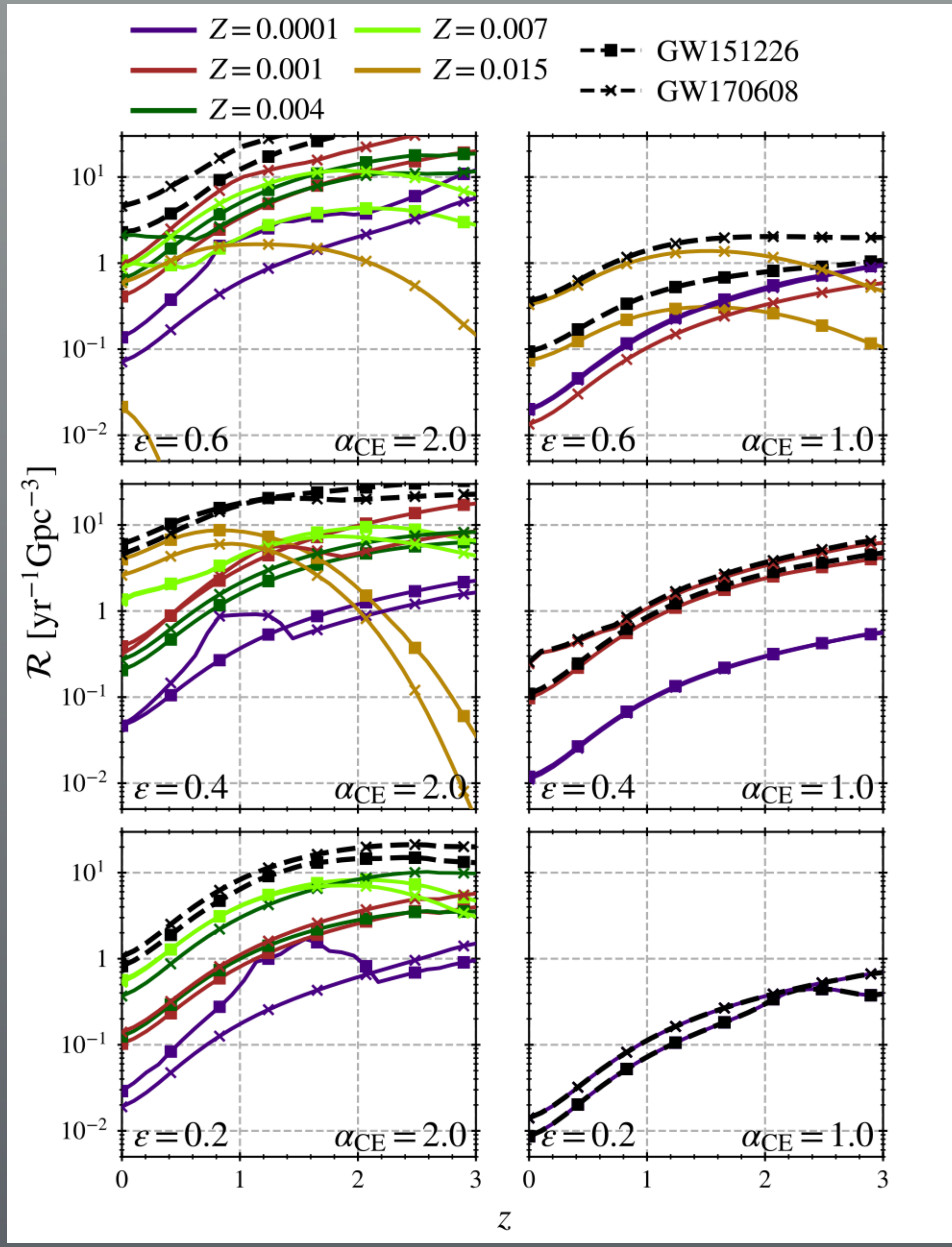
Population synthesis vs binary star evolution

- Simulation of the binary formation and evolution processes: Two approaches
  - Binary Population Synthesis (BPS)
  - Binary star evolution (BSE)

- This talk : BSE
  Aim : simulate a population of binaries with MESA and use machine learning to learn the input/output relation of these simulations



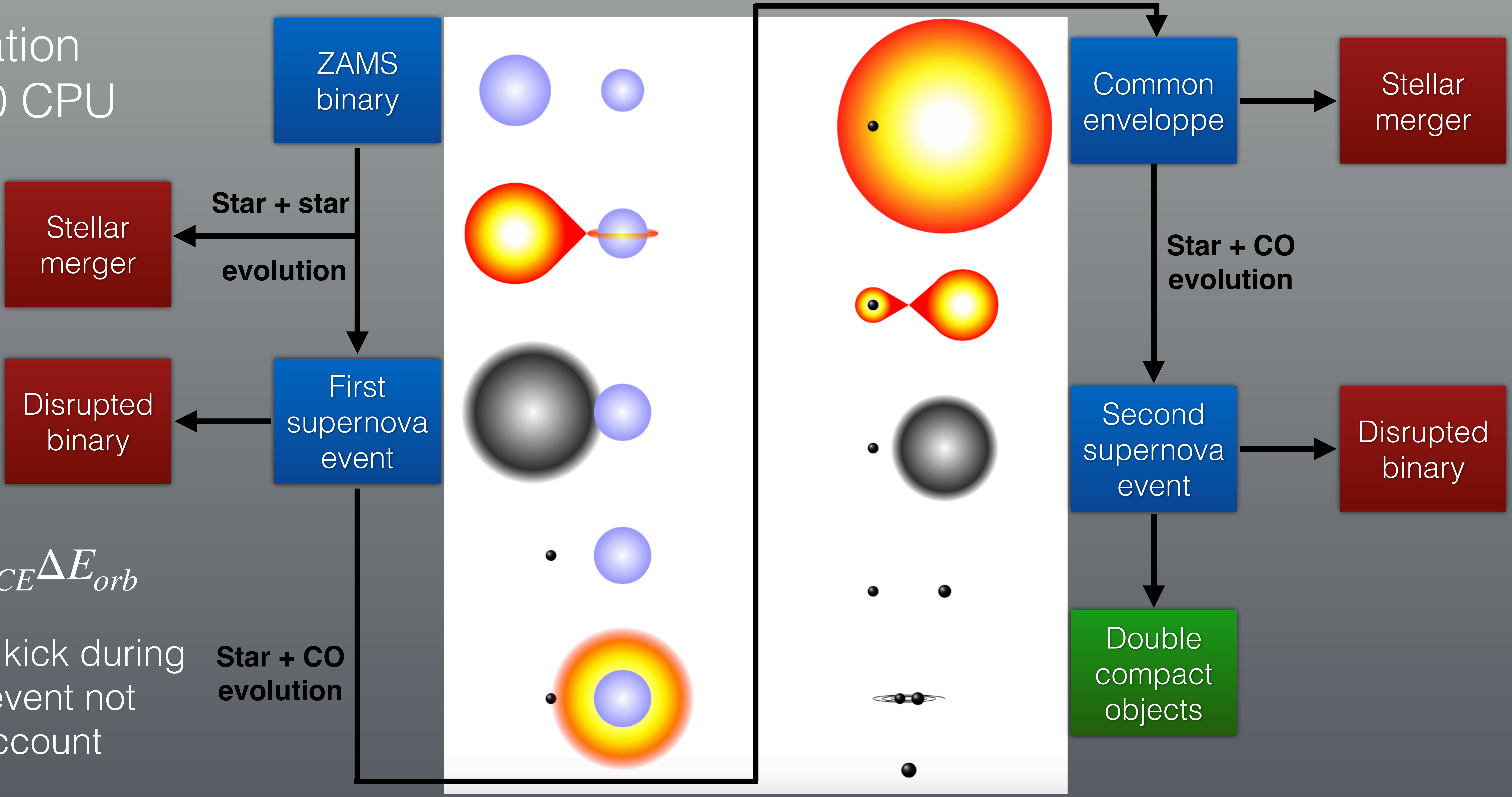
Merger rate density of DCO. Credits Santoliquido et al. (2020)



Merger rate density of DCO. Credits García et al. (2021)

MESA simulations

- 1 simulation ~10-100 CPU hours



- CE phase : $\Delta E_{bind} = \alpha_{CE} \Delta E_{orb}$
- Asymmetric kick during supernova event not taken into account

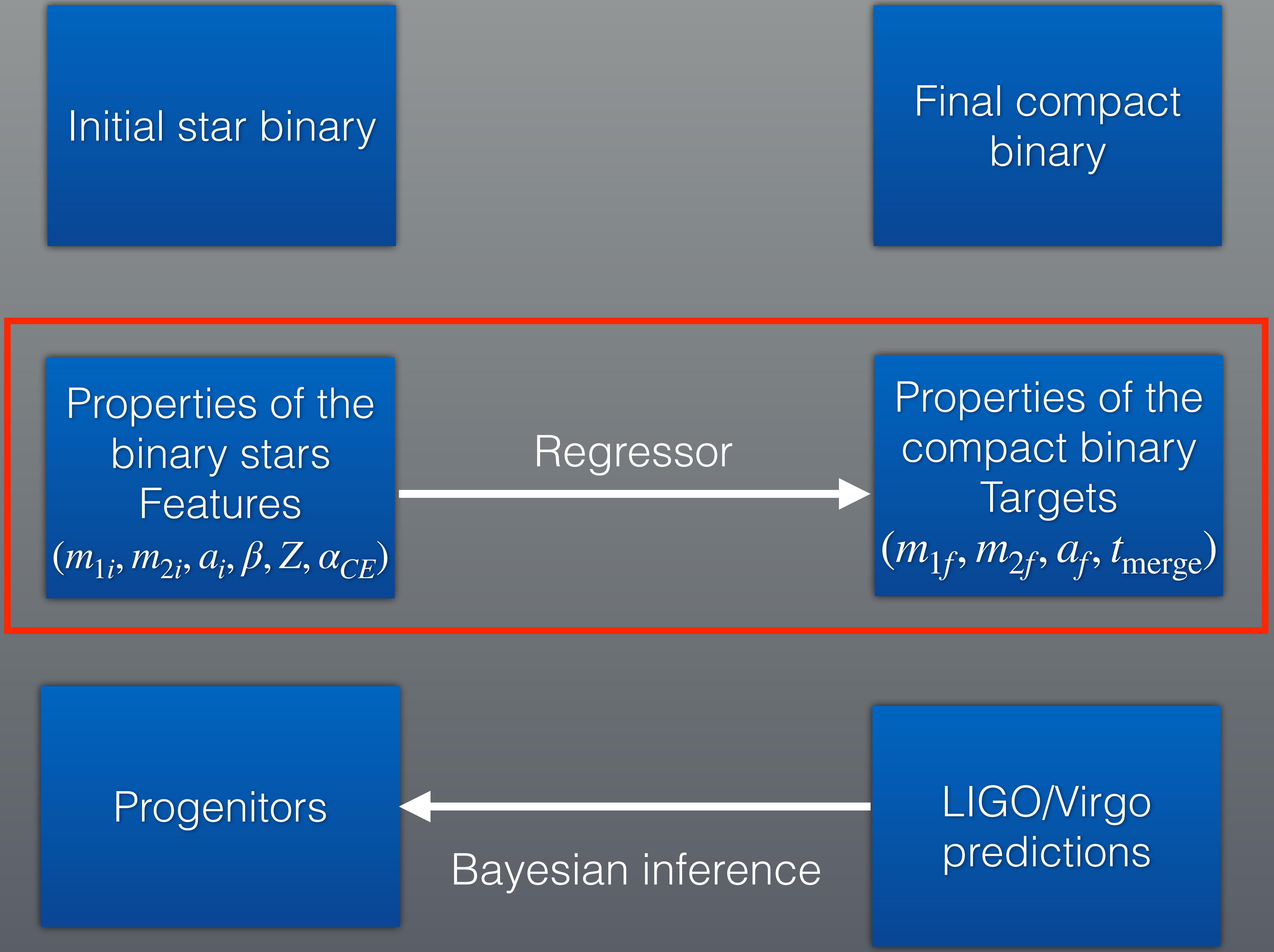
Simulated population of binary star

- 27k simulations in total, 22k from García et al. (2021)
- Final compact binary :
  Mass of the first object : m_{1f}
  Mass of the second object : m_{2f}
  Orbital distance : a_f
  Merging time :t_{merge}

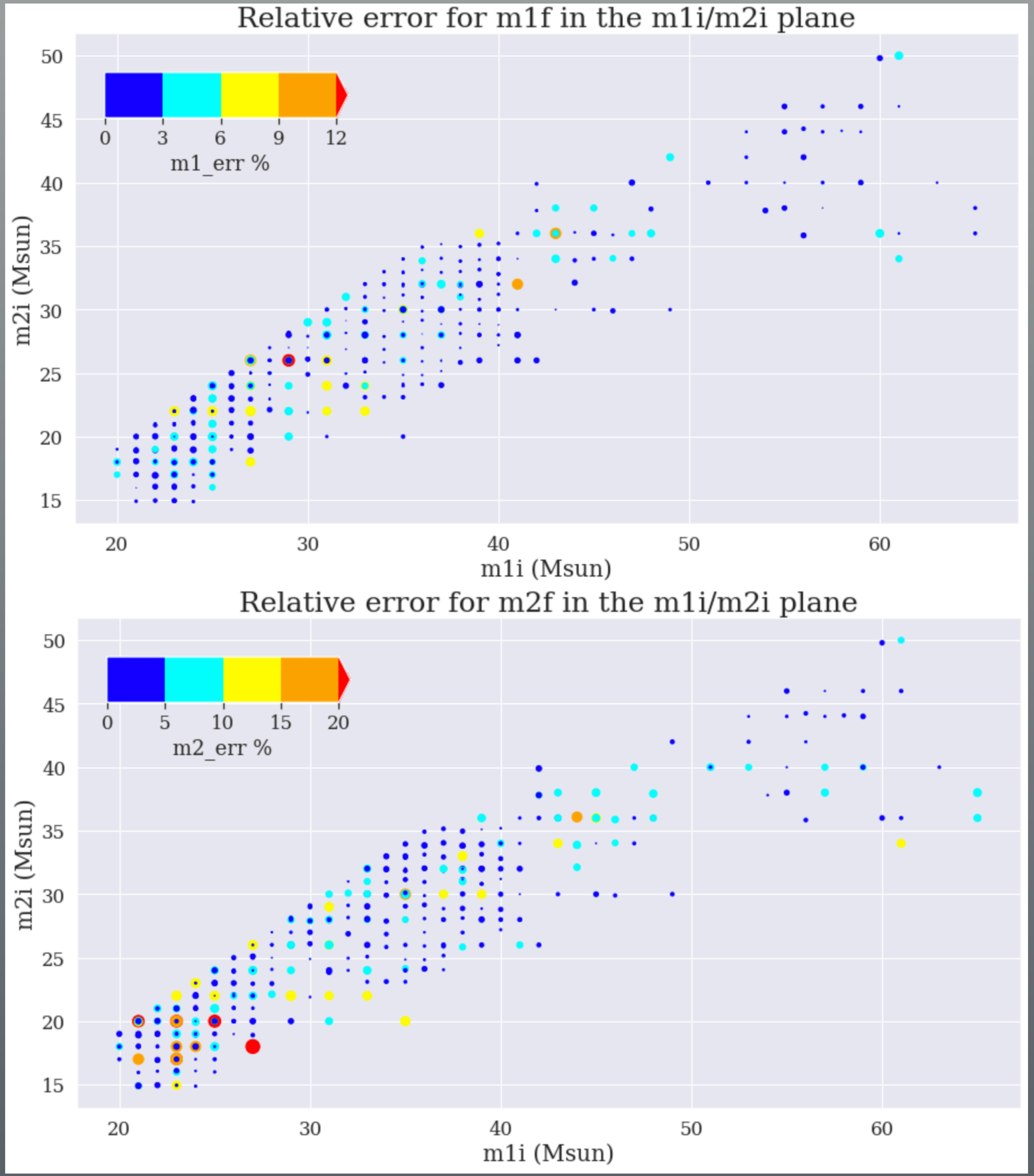
Initial binary	Range value
Mass of the first star m_{1i}	$[20M_\odot,91M_\odot]$
Mass of the first star m_{2i}	$[14M_\odot,62M_\odot]$
Orbital distance a_i	$[30R_\odot,500R_\odot]$
Metallicity $-\log(Z)$	$\{1.8,2.2,2.4,3,4,5\}$
Mass transfer efficiency β	$\{0.2,0.4,0.6,0.8\}$
Common enveloppe efficiency α_{CE}	$\{1,2\}$

Machine Learning project

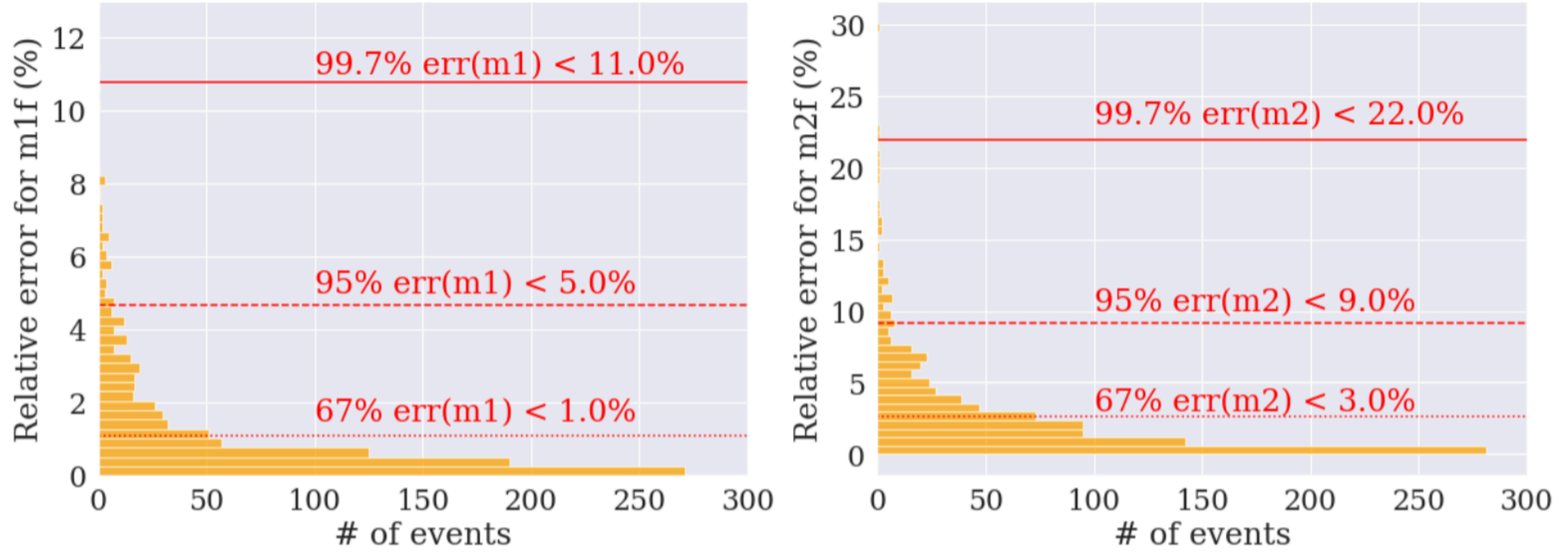
- Use MESA simulations to train regressor to infer final state from initial parameters of massive binary stars
- We want to train it such that error(target) < error(LIGO/Virgo)



Results



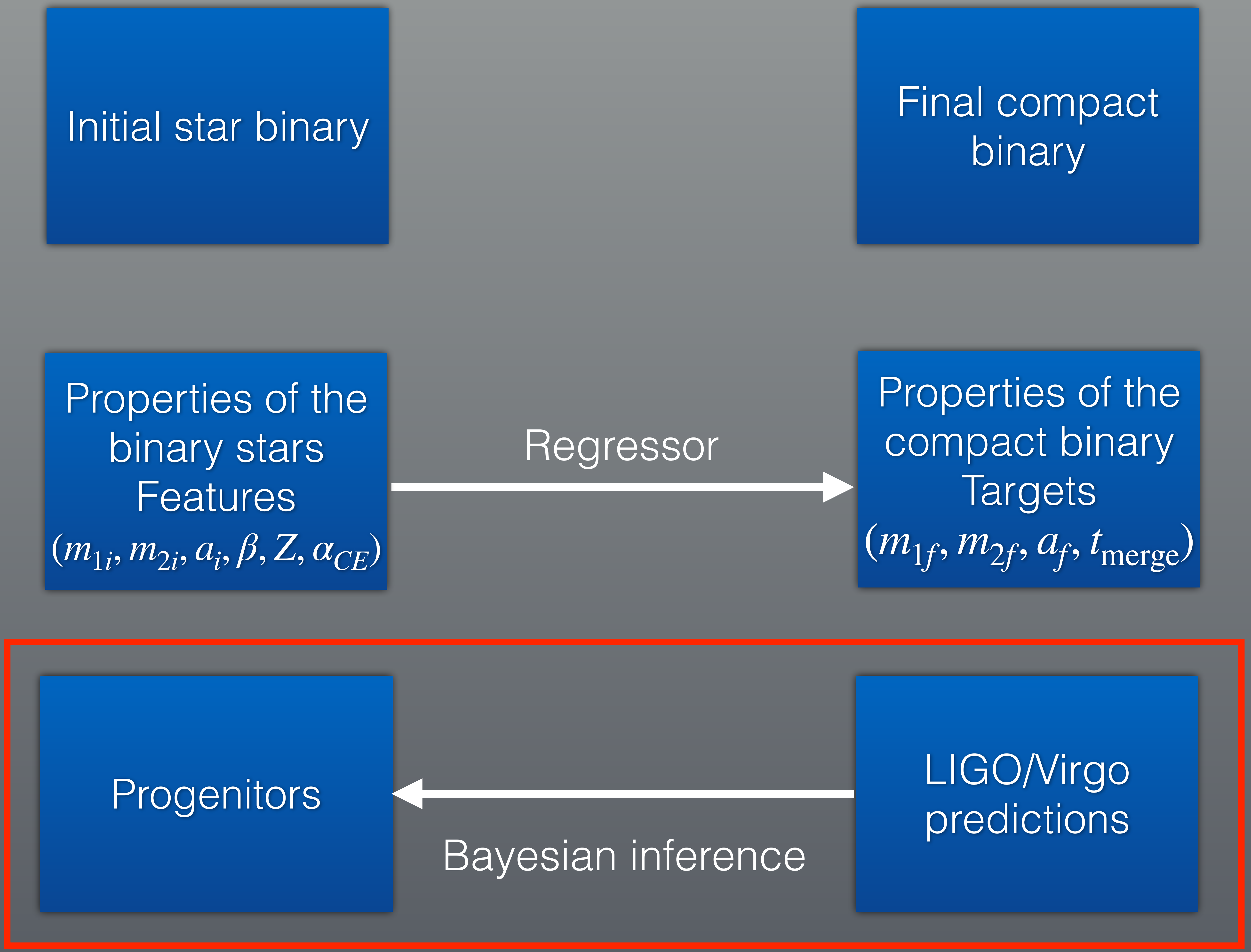
Relative error on final mass 1 (top) and final mass 2 (bottom) in the initial masses plane



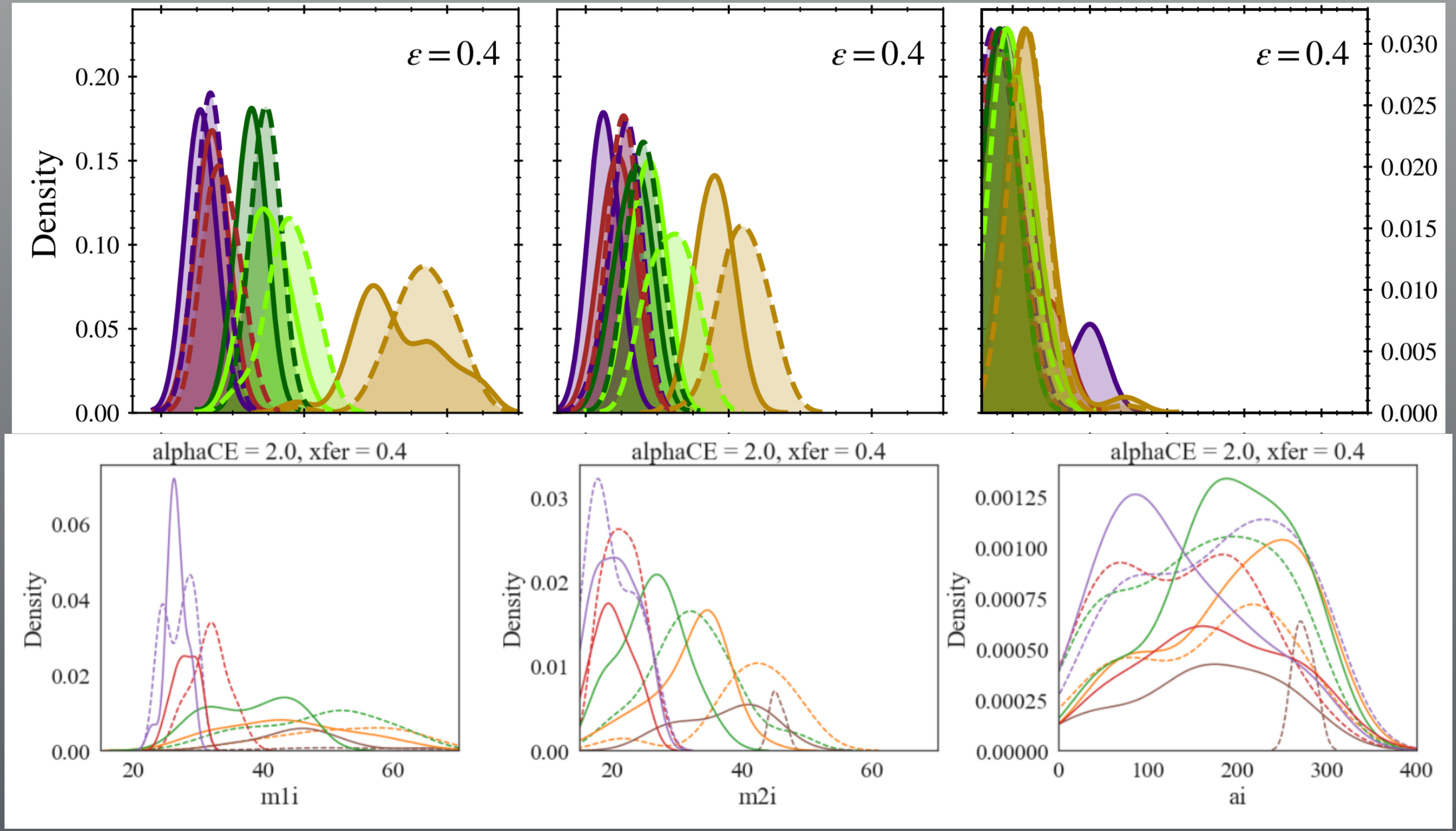
Number of events in the testing dataset with a relative error for predicted final masses within a given bin. Red lines are values where 67% (dotted), 95% (dashed) and 99.5% (solid) of the dataset are below.

Inference of the BBH progenitors

- We can now simulate (regress) the binary evolution very fast
- Use the regressor to find binaries that produce BBH compatible with the LIGO/Virgo posteriors
- Use a Bayesian sampler to do the inverse problem



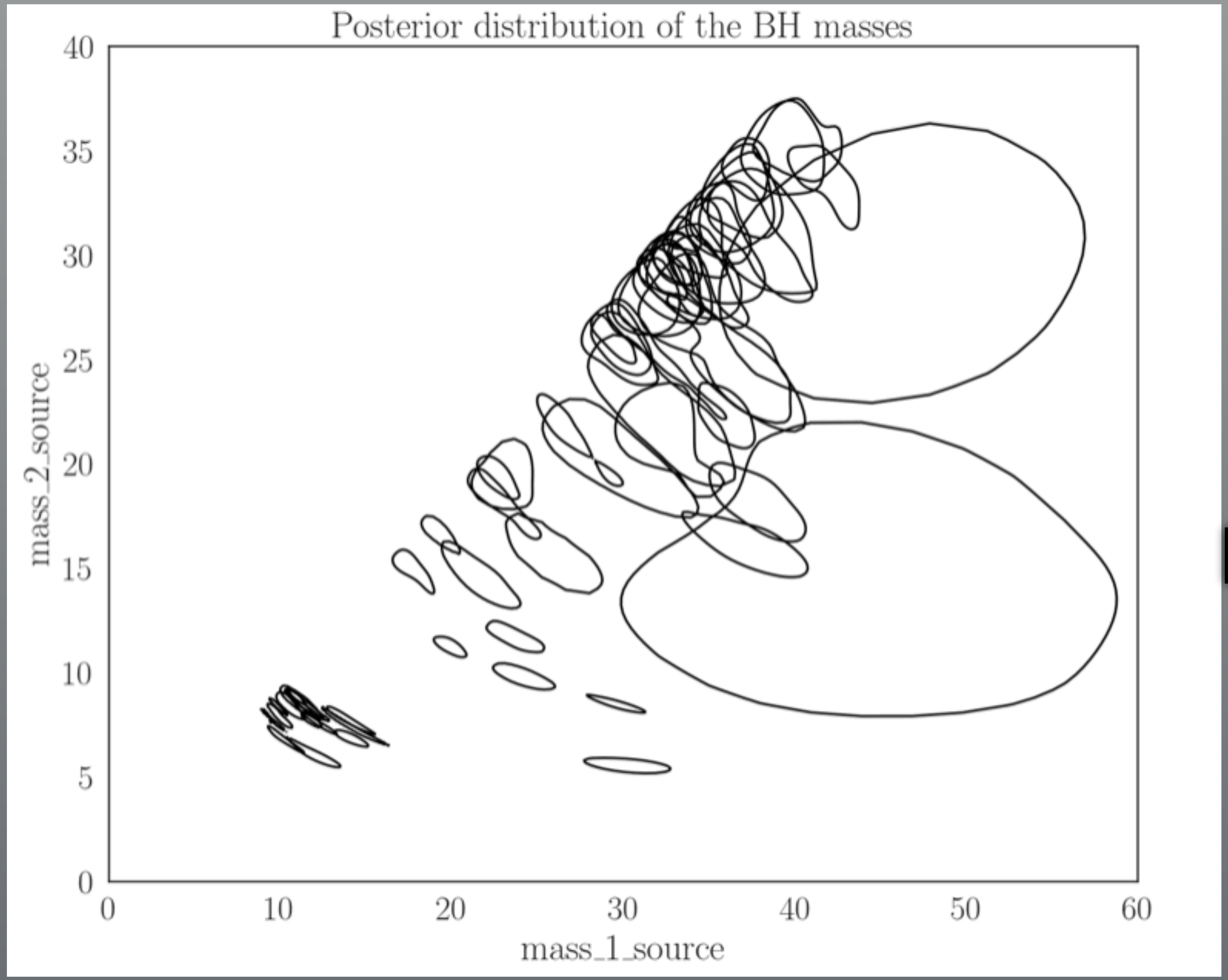
Comparison with the results of Garcia et al, 2021, on GW151226 and GW170608



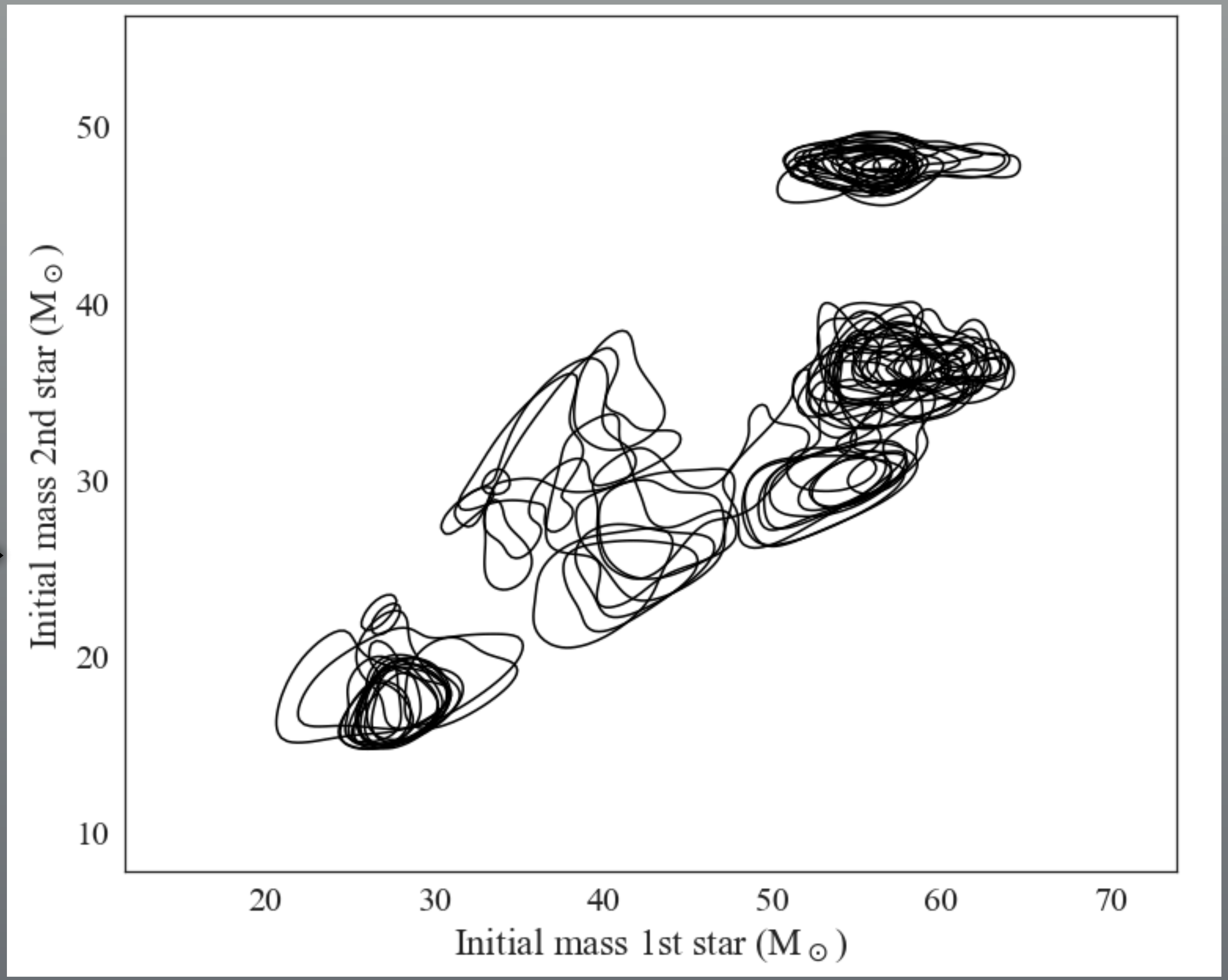
Comparing results from MESA simulations (Garcia et al. 2021) and from Bayesian inference for two GW events (GW151226 and GW170608)

- Initial masses well constrained by the algorithm, in agreement with results from simulations
- However, orbital separation is not in agreement with results from simulations

Results with GWTC-3



Bayesian Inference



Posterior mass distribution of the BH masses

Initial mass distribution of the GW progenitors

- 65 gravitational wave events tested with the algorithm
- First results give a glimpse of the existence of groups of binary systems covering the final phase space
- Need to dig further in the biases brought by the algorithm.

Summary

- Interest of the work :
  - MESA simulations describe with good precision physics behind the evolution of binary stars, giving a useful dataset to study progenitors.
  - ML drastically reduces the computing time of the final state of a binary

- Results :
  - ML based evolution model for black hole binaries in the $m_1 \in [3,30] - m_2 \in [3,40]\, M_\odot$ mass range able to predict component mass with 20% accuracy.
  - Use of Bayesian inference on LIGO/Virgo events mass posterior distribution allow to constrain the initial masses of the progenitors.

- Limits :
  - We are in a fixed phase space, expanding it is computationally expensive
  - ML does not perform well at predicting final orbital separation and merging time.
  - Bayesian inference not with agreement with former studies on orbital separation

BACK UP

ML

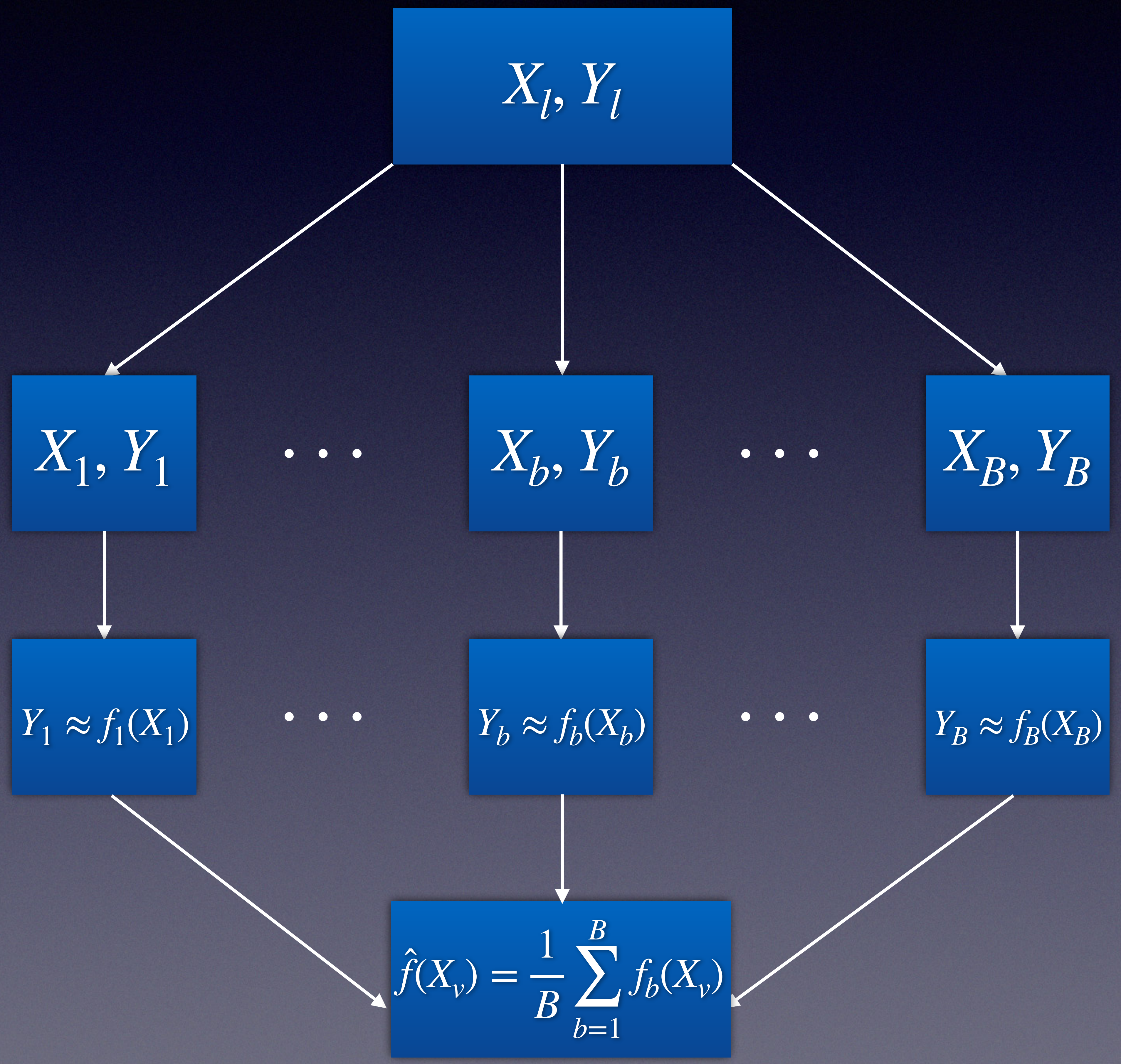
Machine Learning project

The algorithm :

$$\text{Learning set} : X_l = \{x_{l,i}\}_{i\in[1,n]}, Y_l = \{x_{l,i}\}_{i\in[1,n]}$$

$$\text{Validation set} : X_v = \{x_{v,i}\}_{i\in[1,n]}, Y_v = \{x_{v,i}\}_{i\in[1,n]}$$

A RFR algorithm with B trees will simulate B new datasets $(X_b, Y_b)_{b\in[1,B]}$ of size n, using selection with replacement from the original dataset, and apply regression tree on the B new datasets.

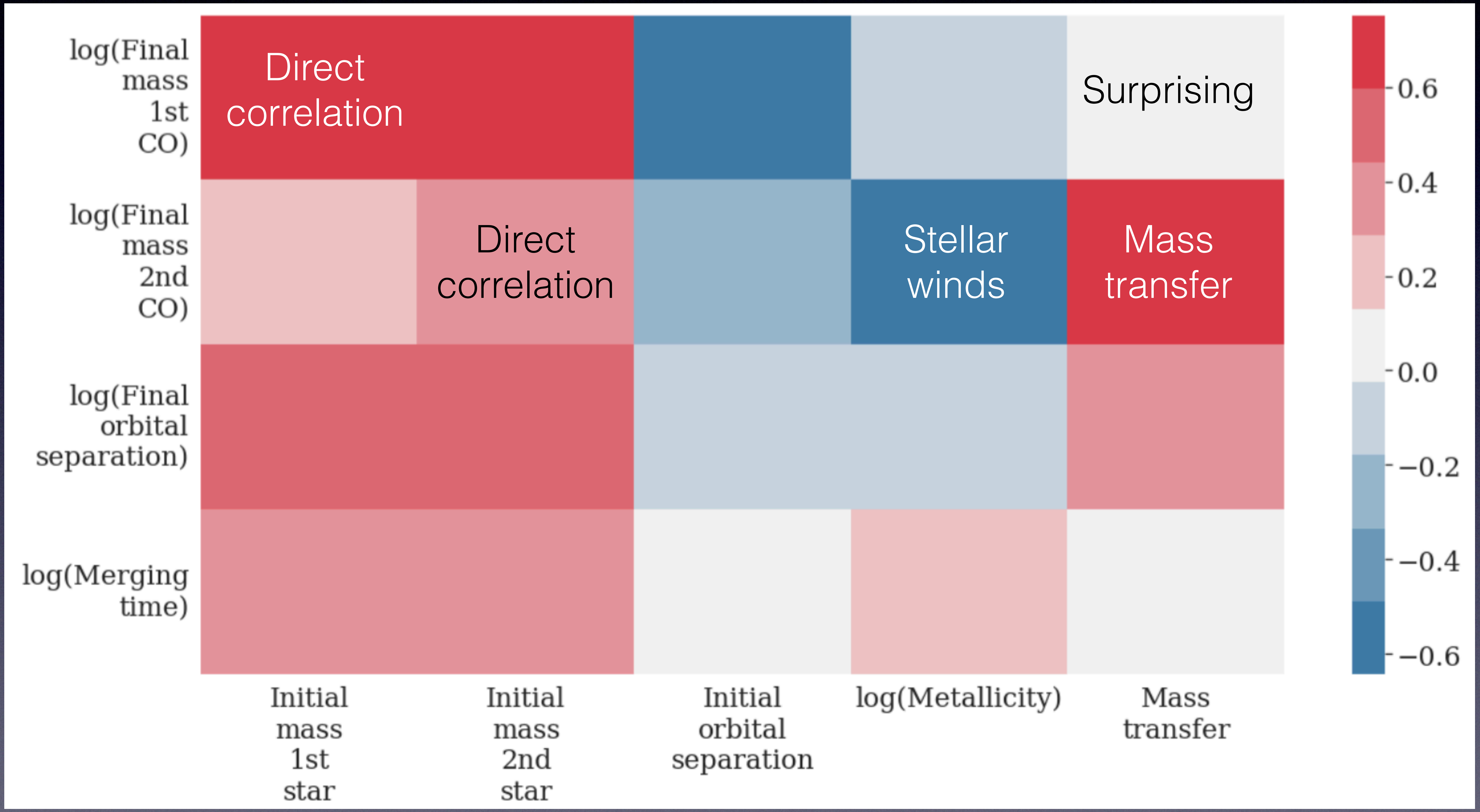


Schematic description of a Random Forest Regressor

Results (1) : Sanity check

- Strong correlations between final masses and initial parameters

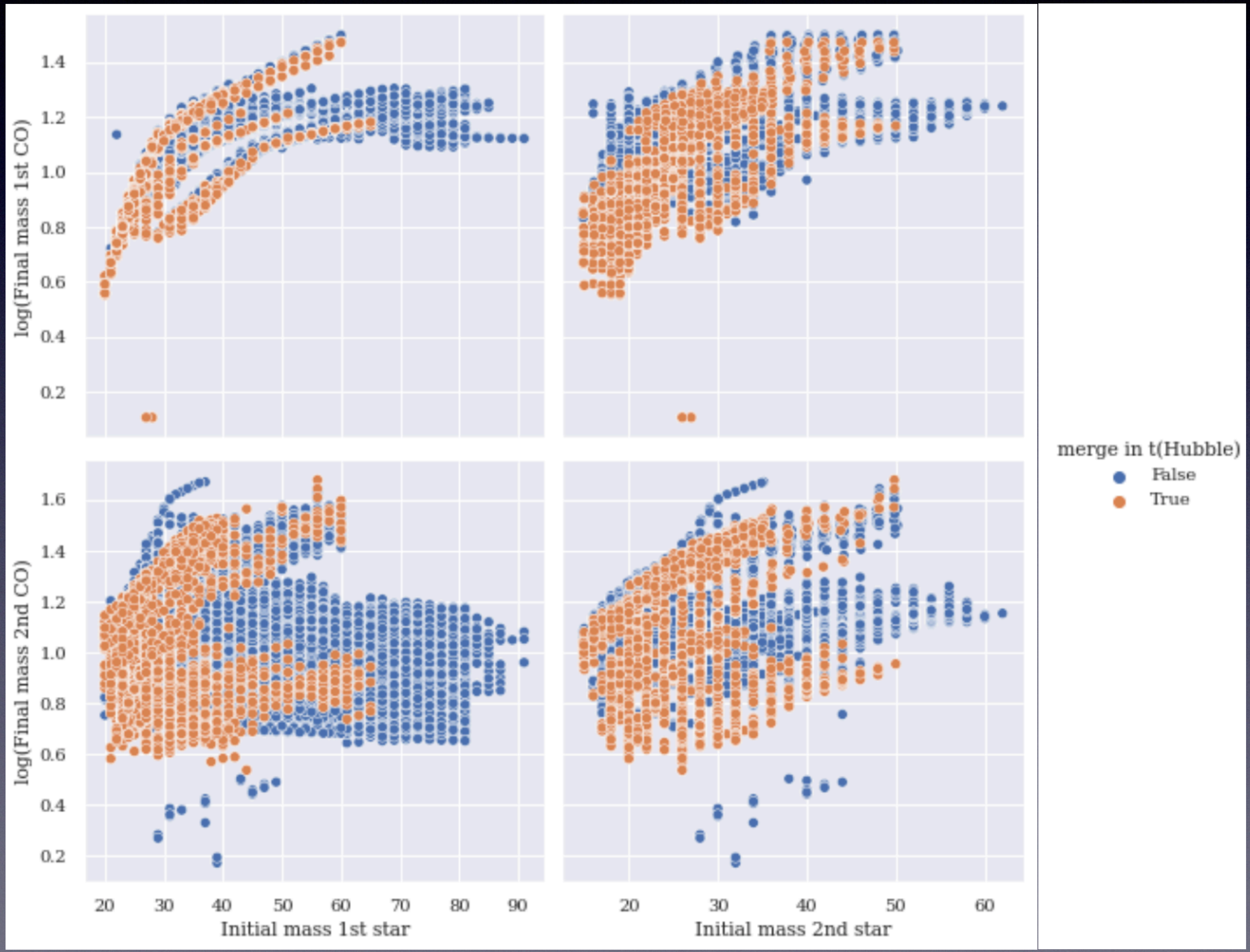
- Weaker correlations for final orbital separation and merging time



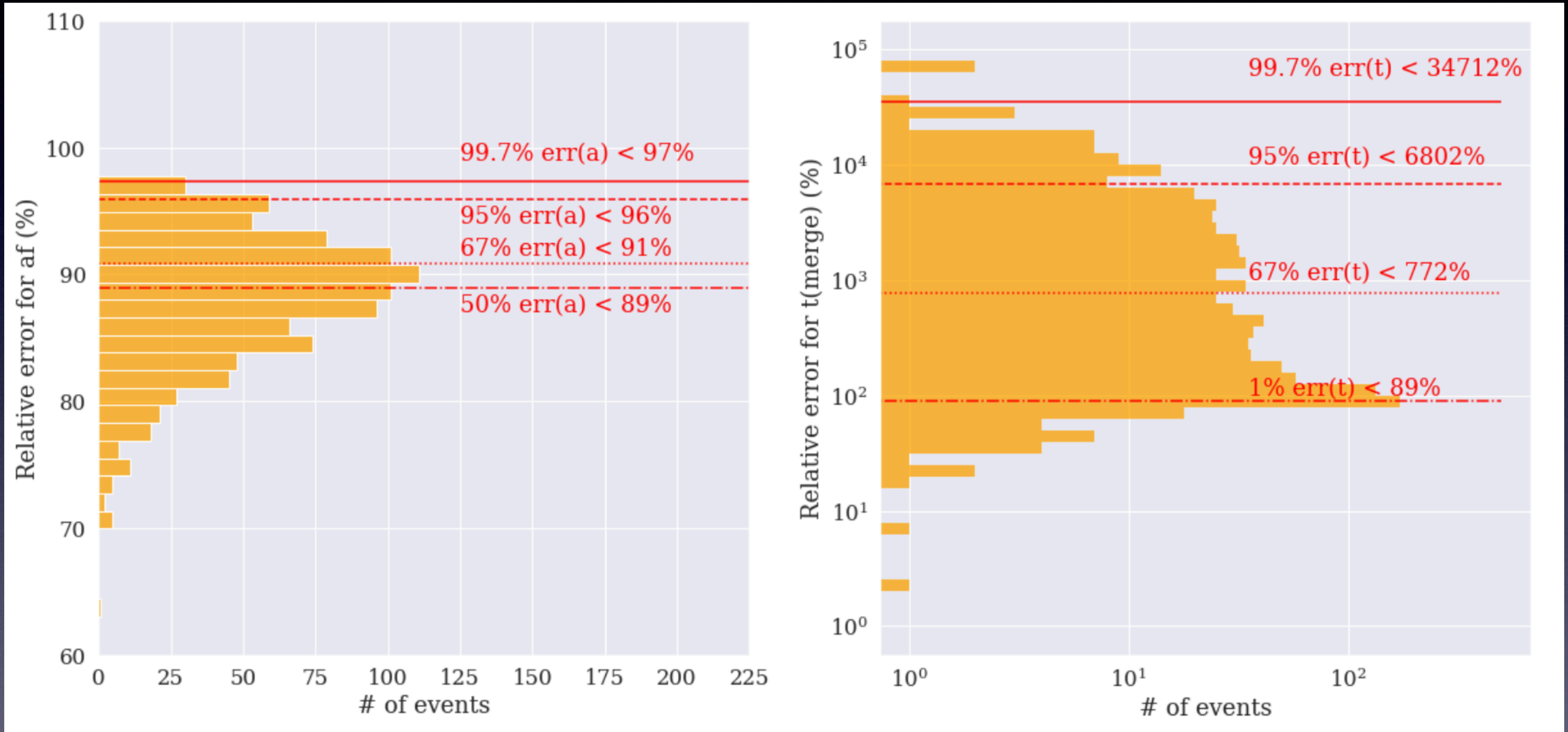
Correlation map between initial and final parameters

Merger events in the dataset

Termination	Number	Percentage of the total dataset
CE merge phase	3157	12 %
Binary disruption	33	< 1%
Numerical issues	2733	10 %
Mergers (within Hubble time)	21170 (3197)	78 % (12 %)
Total	27093	100 %

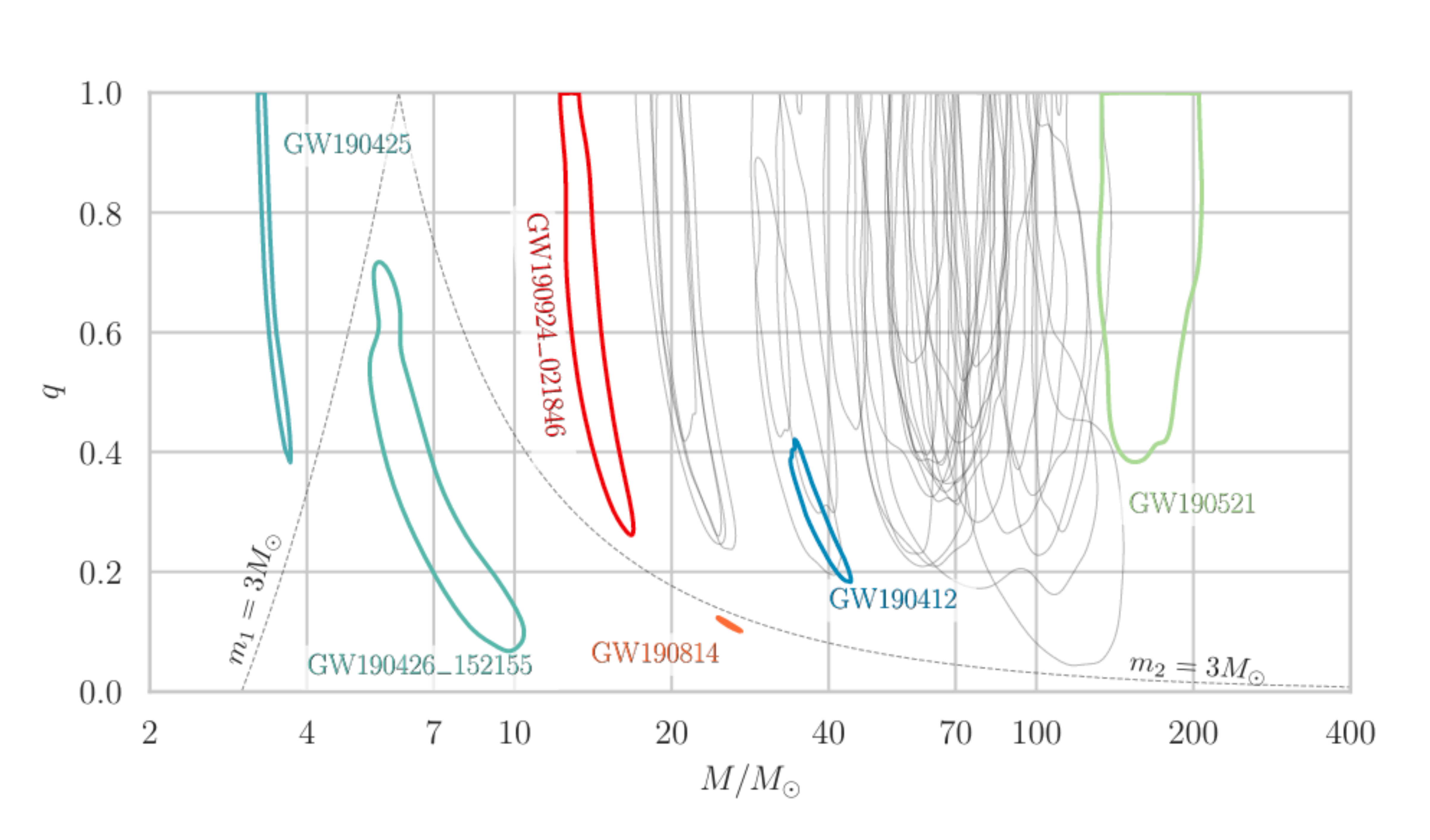


Representation of the dataset

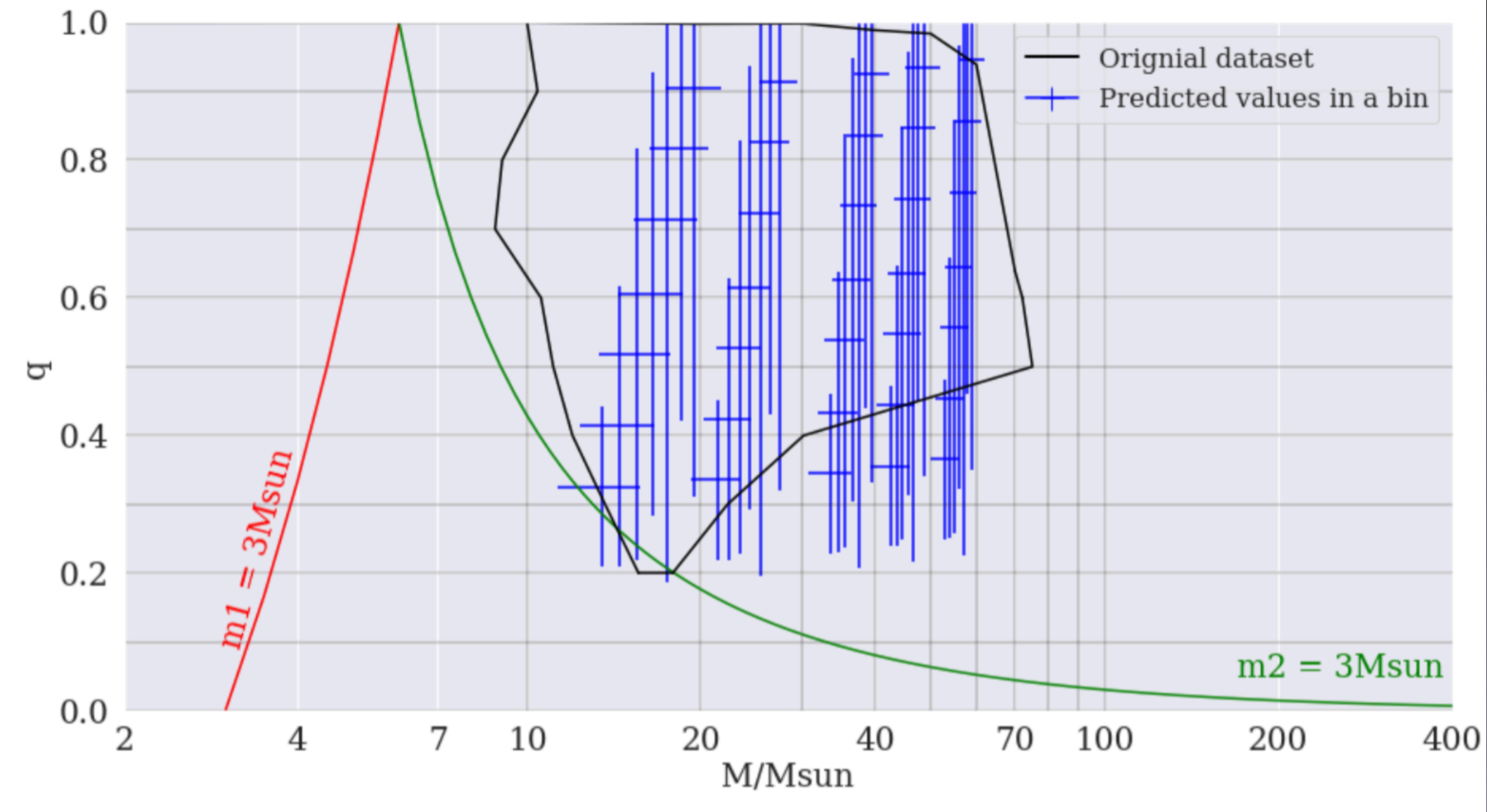


Number of events in the testing dataset with a relative error for predicted final orbital separation and merging time within a given bin. Red lines are values where 67% (dotted), 95% (dashed) and 99.5% (solid) of the dataset are below. Dashdot correspond to 50% (left), and 1% (right)

Results (2)

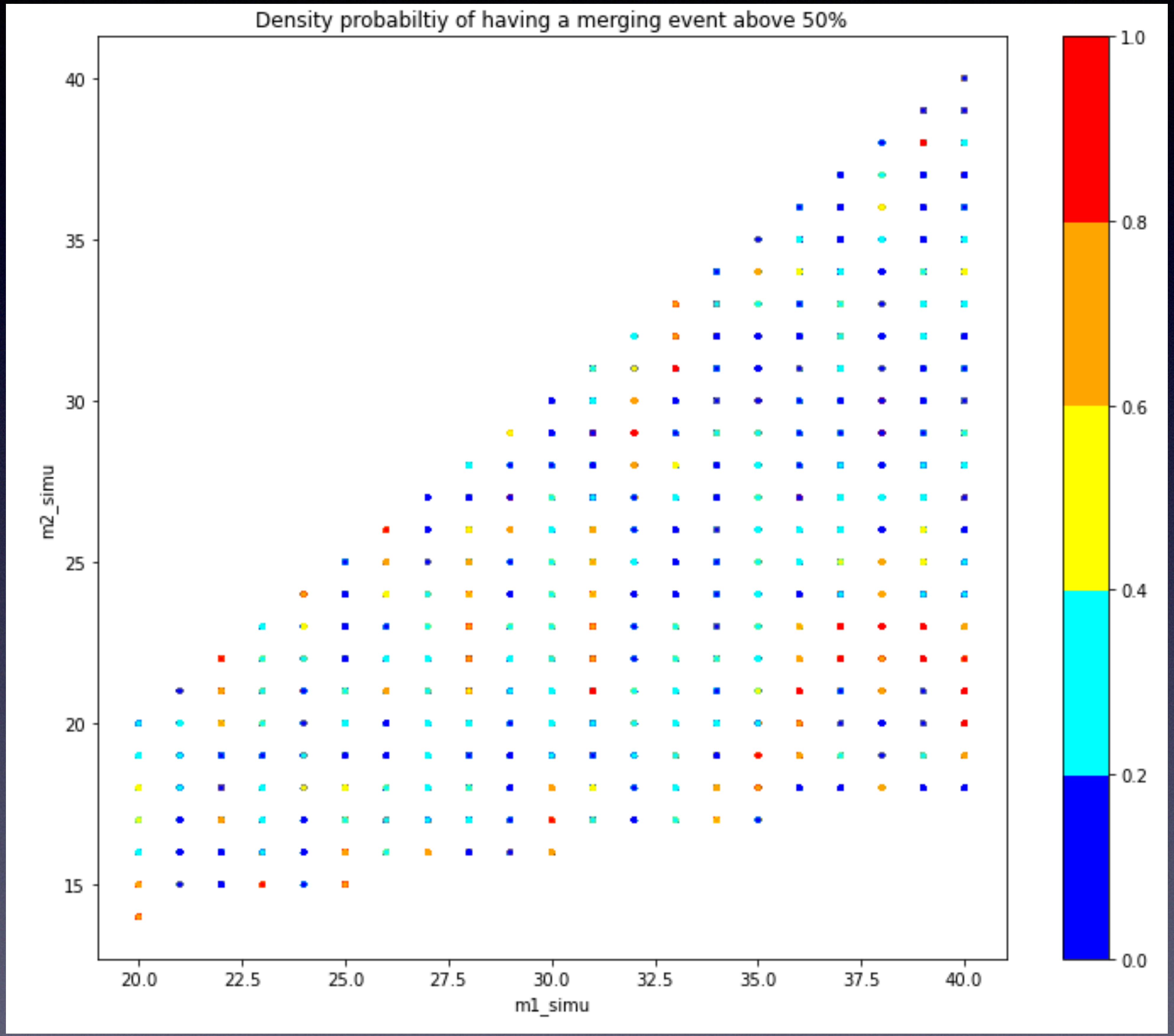


Total mass vs mass ratio of GWTC-2 detections Credits Abbott et al. (2021)

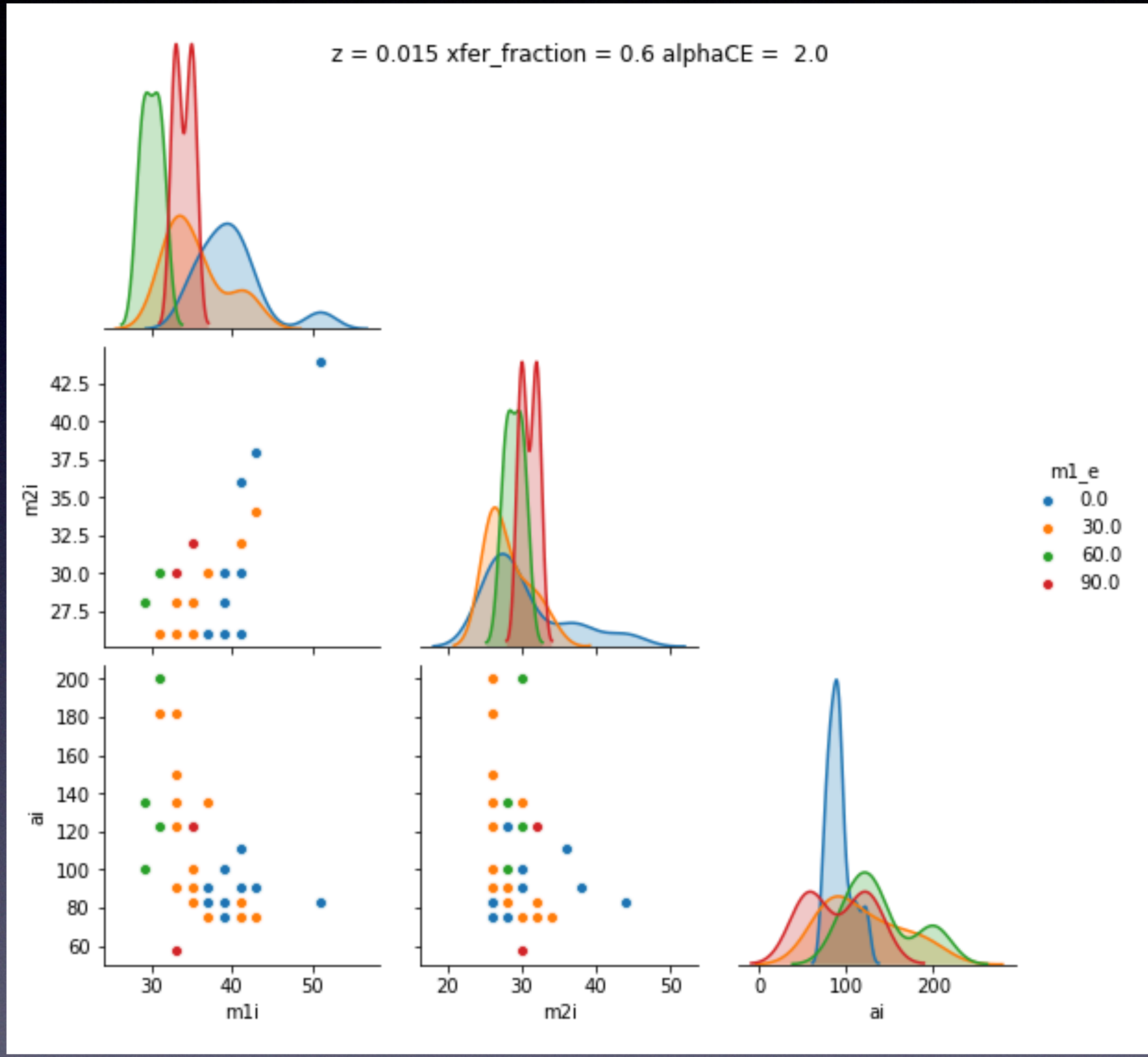


Total mass vs mass ratio for the tested dataset.

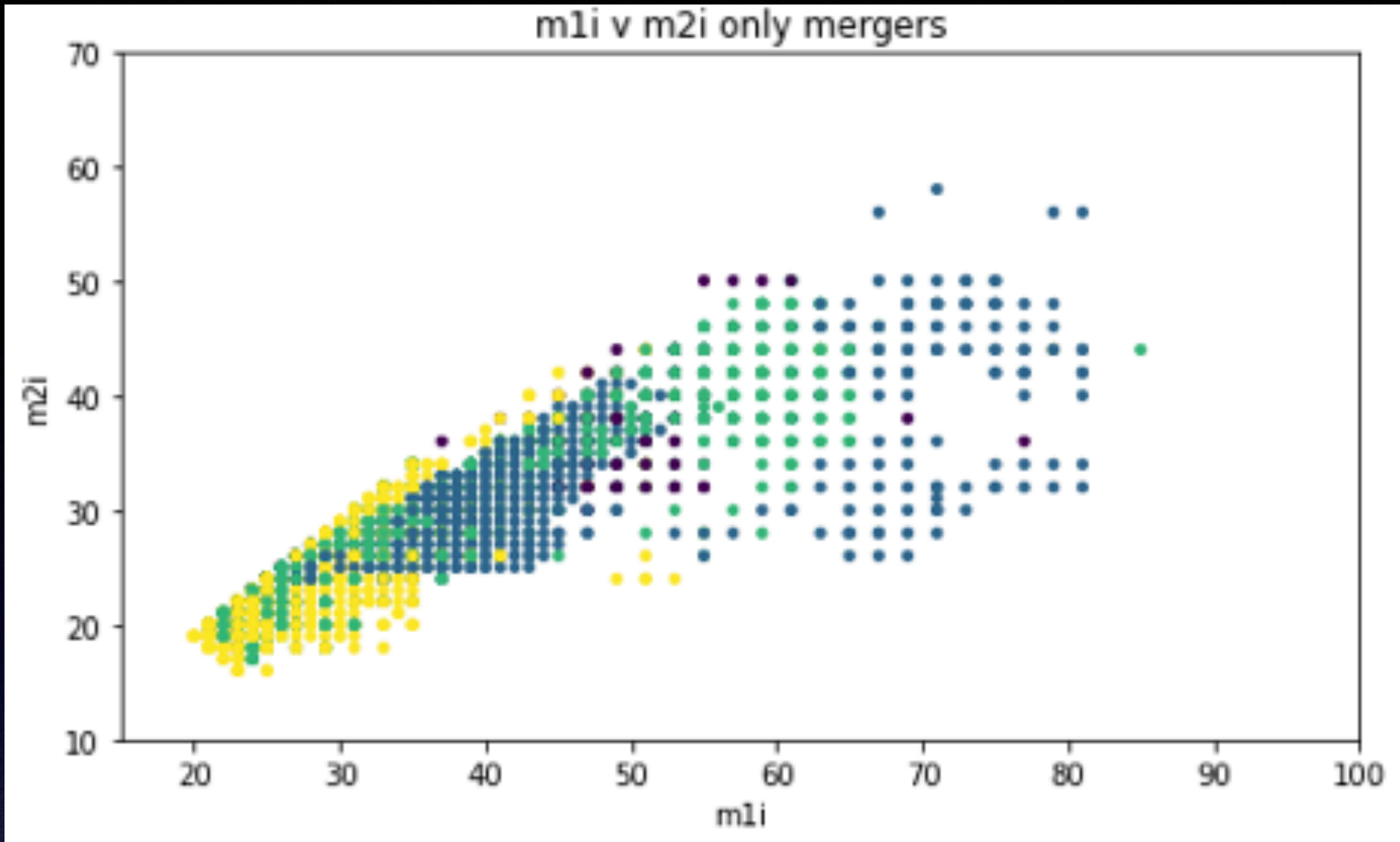
Precision of the model and holes

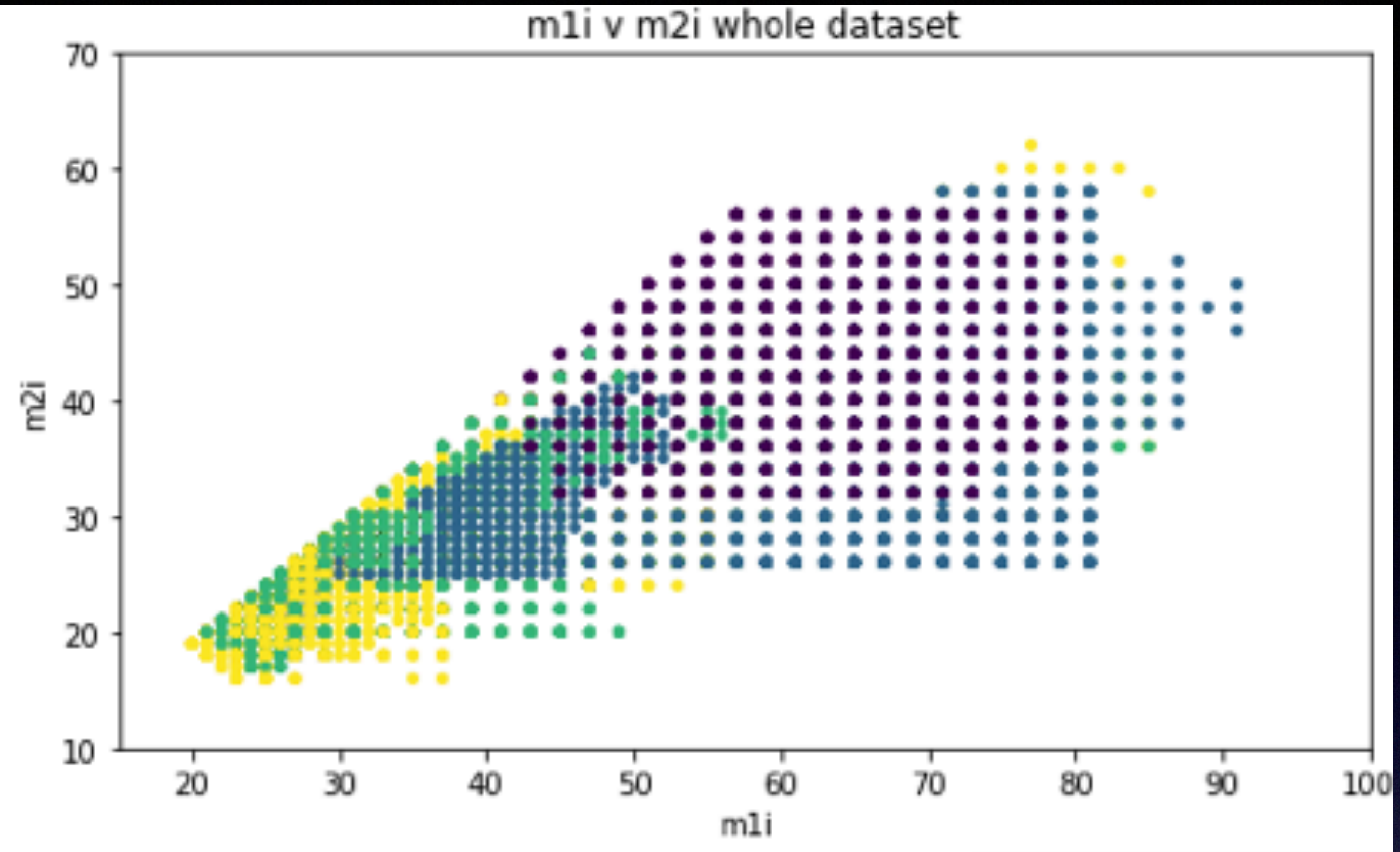


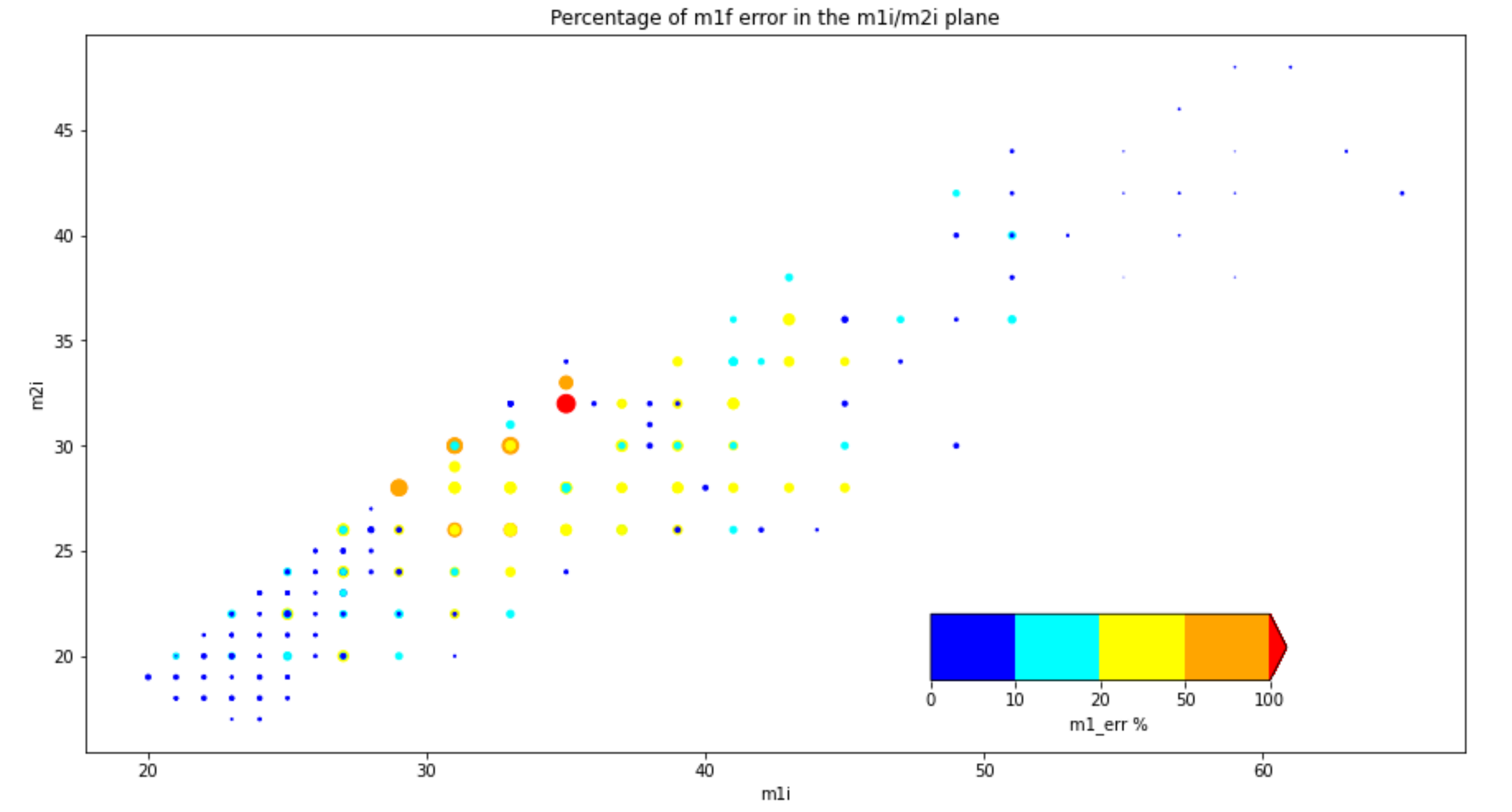
Probability of having a merger in M1i/M2i plane



Representation of a cut in the phase space.
(z=0.015, β=0.4, αCE=2)

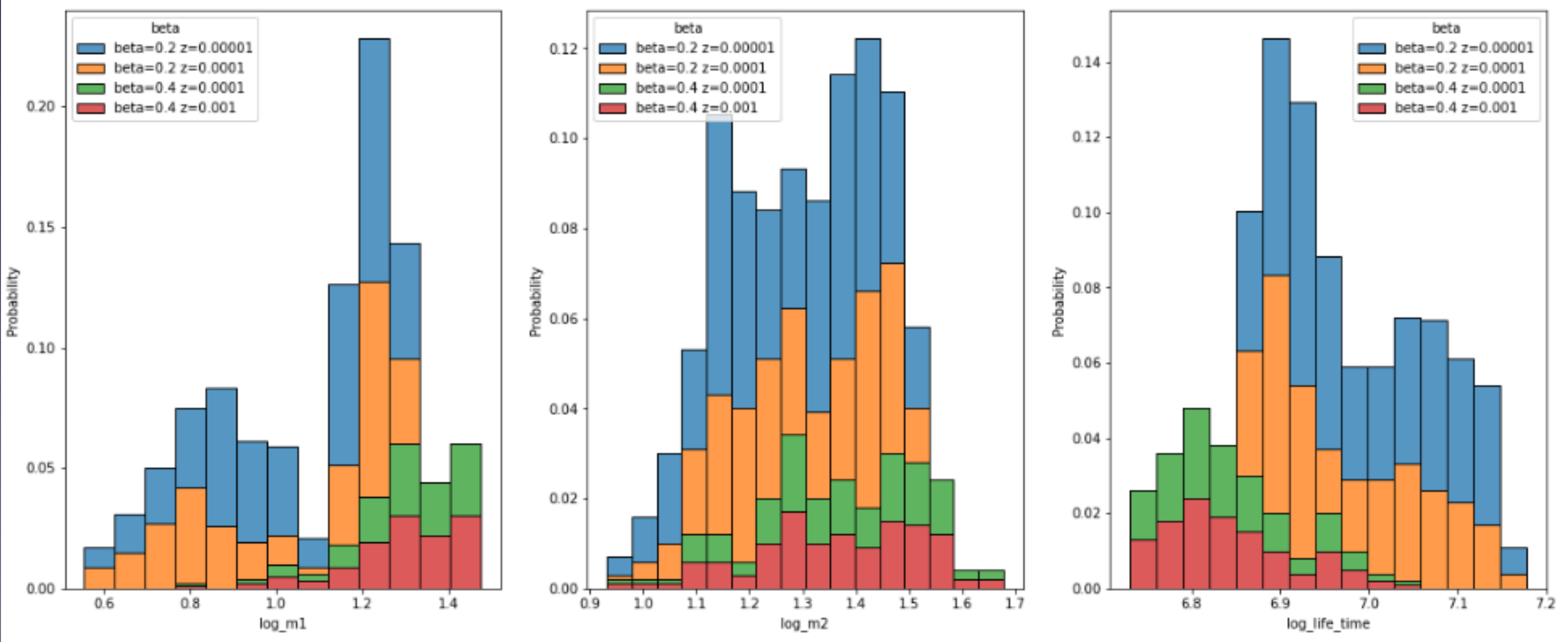

m1i v m2i only mergers
m2i
m1i


m1i v m2i whole dataset
m2i
m1i

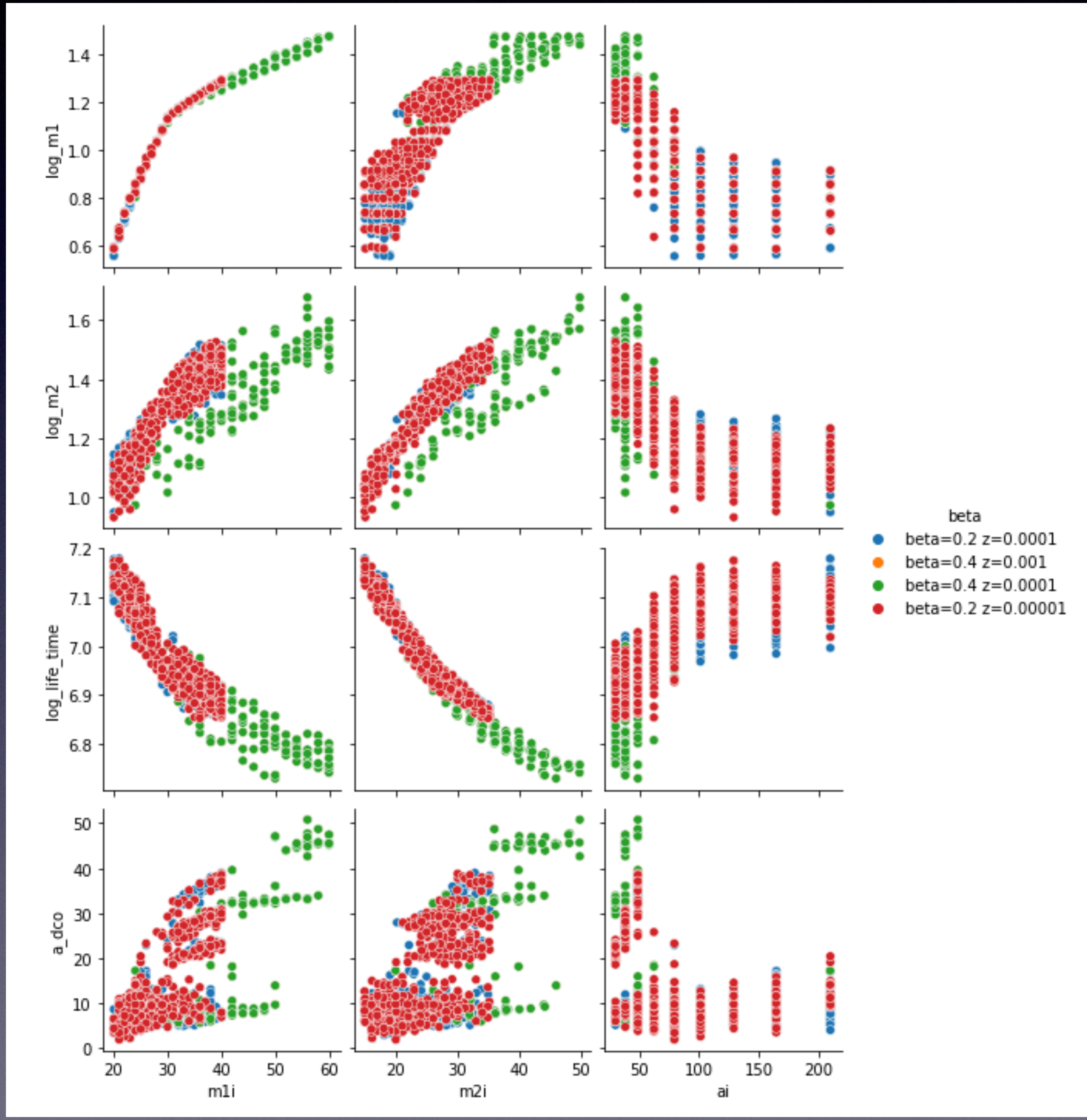

Percentage of m1f error in the m1i/m2i plane
m2i
m1i
m1_err %

Posterior simulations

- 4984 have been done 998 mergers (within Hubble time) have been obtained
- Different z and β tested $(\log(z) \in \{-2\,;-3;-4\};\ \beta \in \{0.2\,;0.4\})$



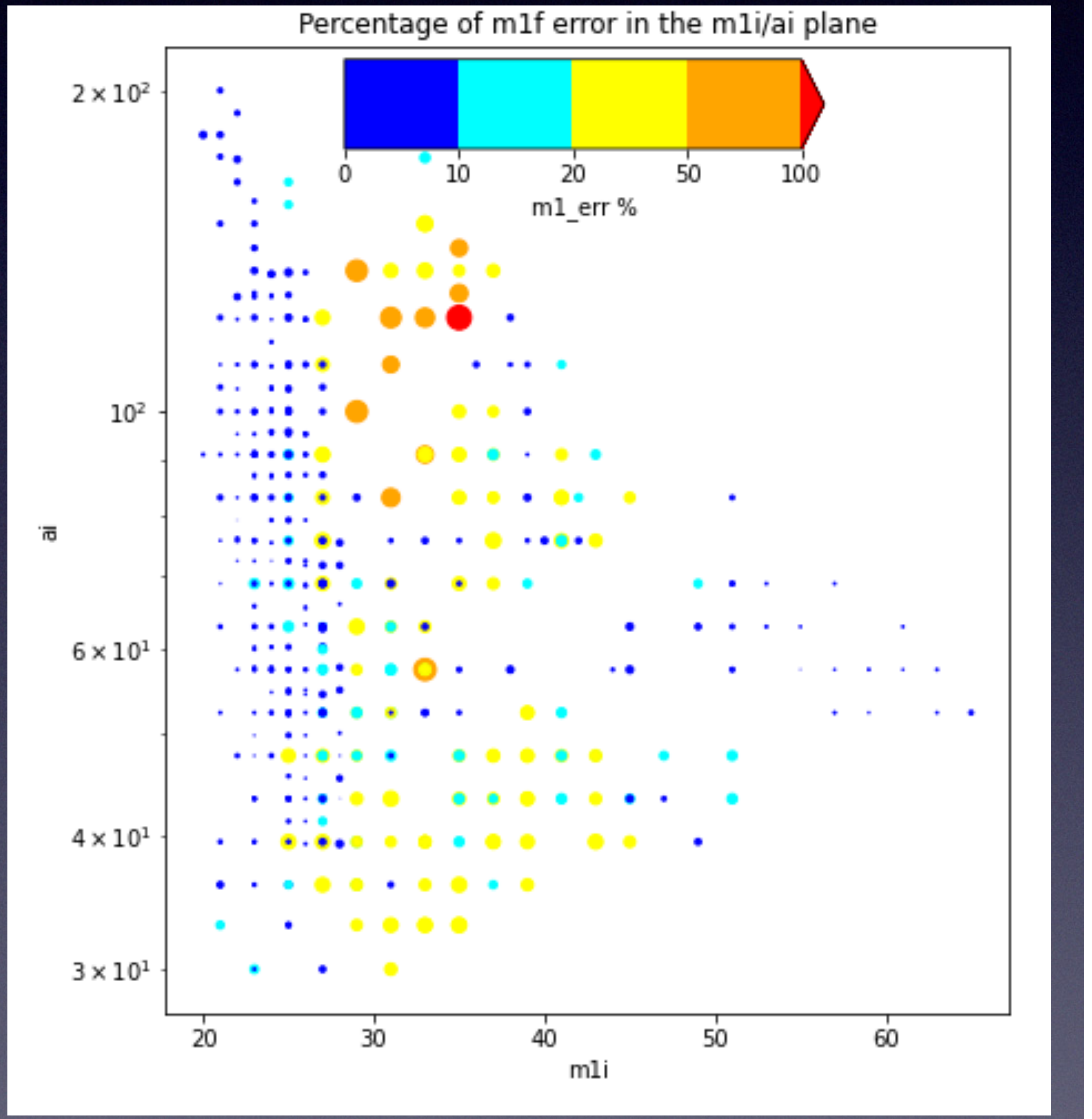
Final parameters histogram repartition. Color indicate β and z.



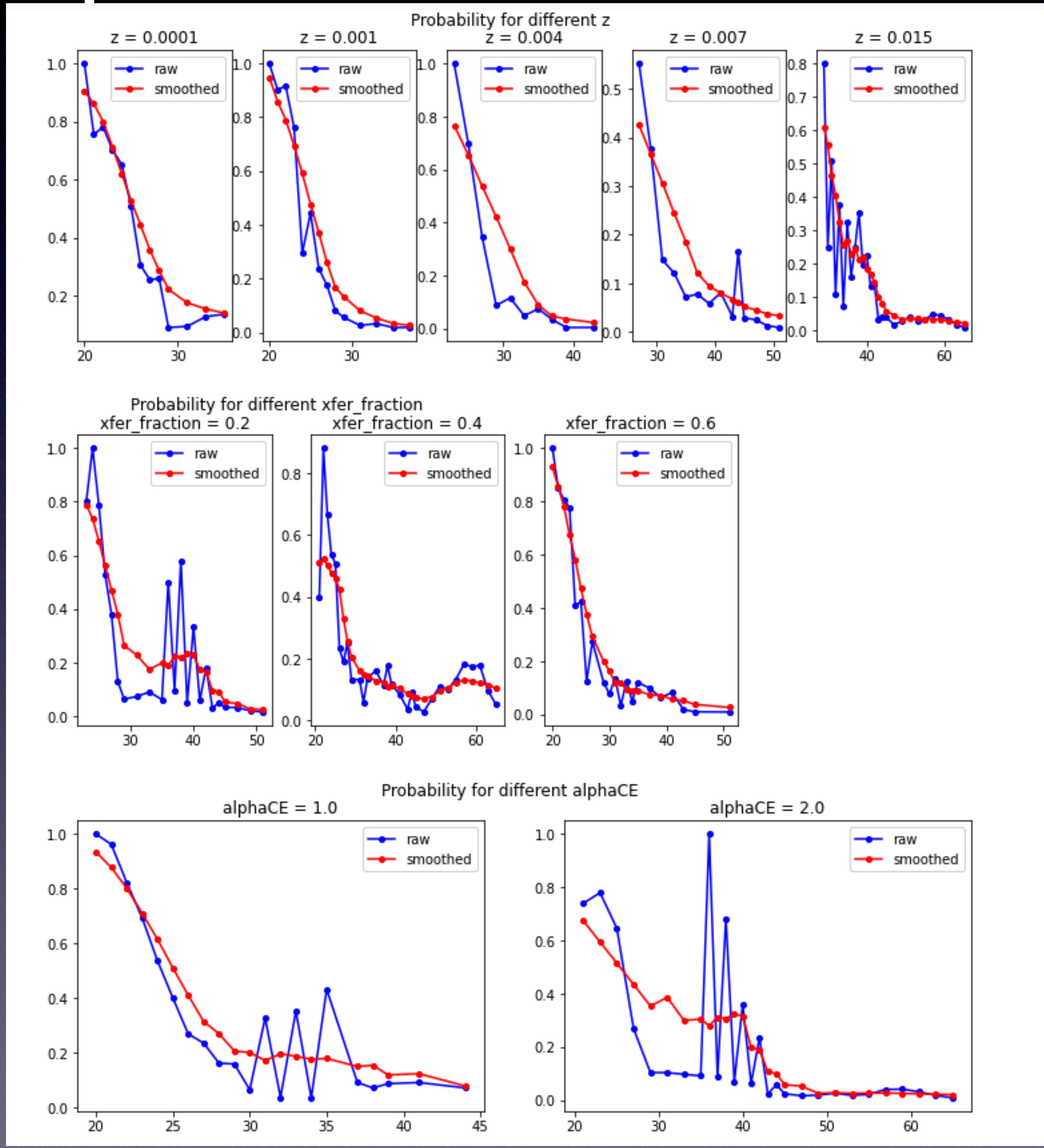
Initial vs final parameters for the simulations. Colors represent β and z

Changing the strat, populate smart

- High computational cost
- Some areas have just holes that have to be filled



Showing holes in the phase space



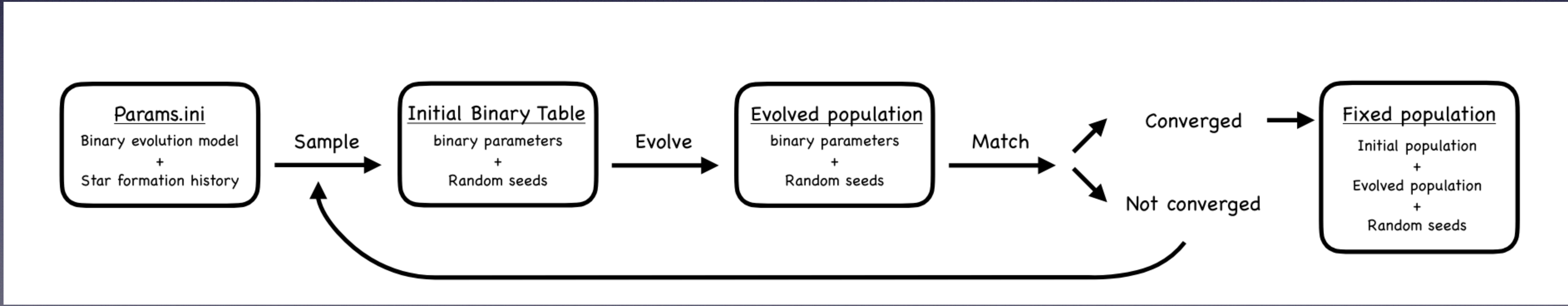
Some regions are less populated, leading to those oscillations in the merging rate density. (Red is what more simulations would look like)

Codes

State of the art in massive binary simulations and inferences on progenitors

Binary Population Synthesis (BPS)

- BPS is a great tool to get a huge amount of binaries, spanning wide ranges of masses and separation in a reasonable computing time.
- Parametric BPS (pBPS) use fitting formulae or look-up tables, but imply strong approximations for the binary systems.

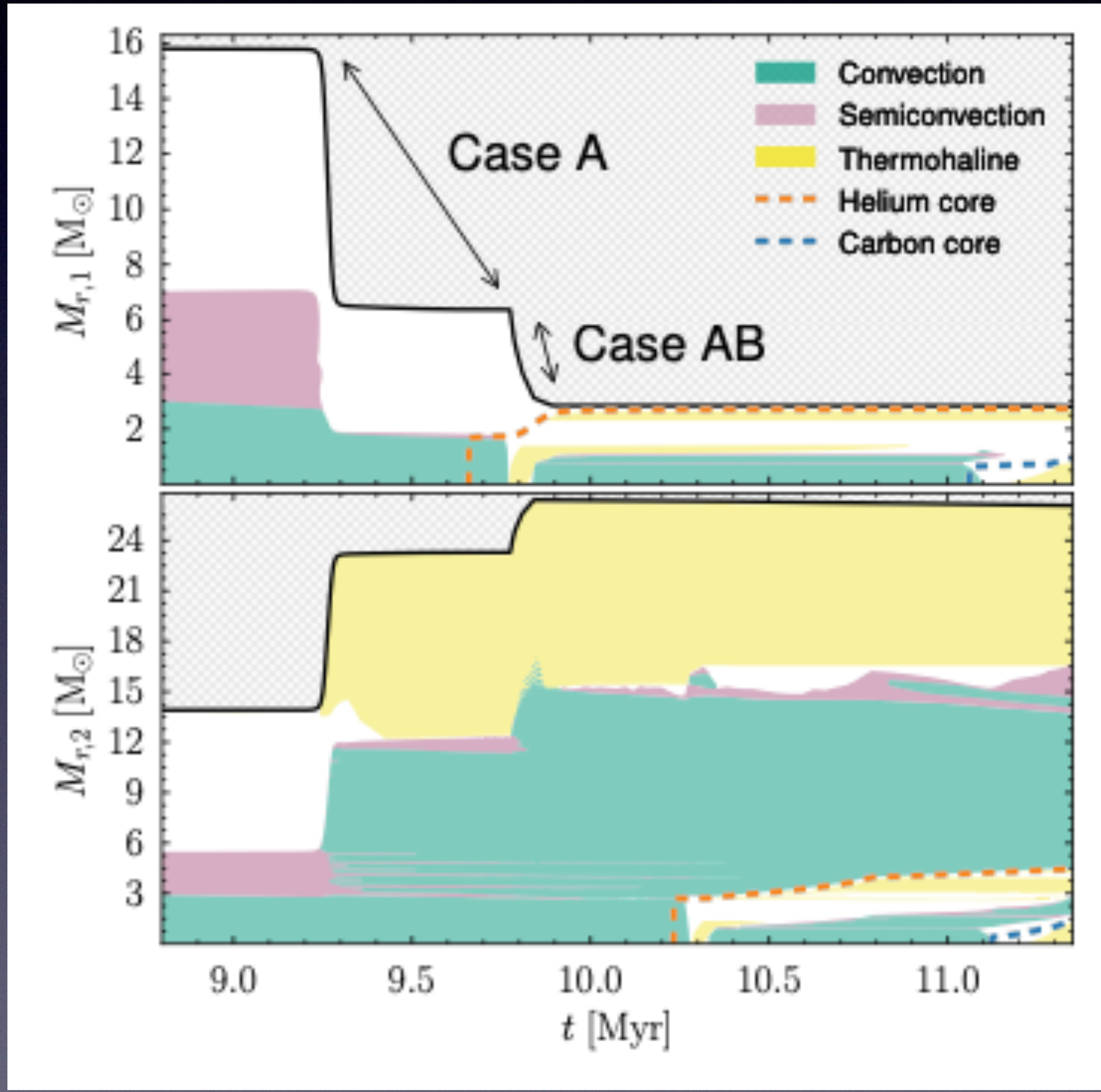


Schematic for the COSMIC BPS code. The evolution is made through a look-up table. Credits : Breivik et al. (2020)

State of the art in massive binary simulations and inferences on progenitors

Detailed Binary Evolution (dBE)

- Detailed Binary Evolution codes are much more time consuming with ~10 — 100 CPU hours per simulation (Paxton et al. 2019).
- They account for more precise physics of the binary and are easily customizable.



Kippenham diagram of a 14 + 16 binary with a 3 days orbital period. Credits Paxton et al. (2017)

BPS

- IBiS (Tutukov & Yungelson 1996, and references therein)
- Brussels' code (Vanbeveren et al. 1998a,b)
- Scenario Machine (Lipunov et al. 1996, 2009)
- SeBa (Portegies Zwart & Verbunt 1996; Toonen et al. 2012)
- BSE (Hurley et al. 2002)
- StarTrack (Belczynski et al. 2002, 2008)
- PNS (De Donder & Vanbeveren 2004)
- binary c (Izzard et al. 2004, 2006, 2009)
- SEVN (Spera et al. 2015)
- TRES (Toonen et al. 2016)
- BPASS (Eldridge & Stanway 2016; Stanway et al. 2016; Eldridge et al. 2017; Stanway & Eldridge 2018)
- COMPAS (Stevenson et al. 2017; Riley et al. 2021)
- ComBinE (Kruckow et al. 2018)
- COSMIC (Breivik et al. 2020)
- MOBSE (Giacobbo et al. 2018)

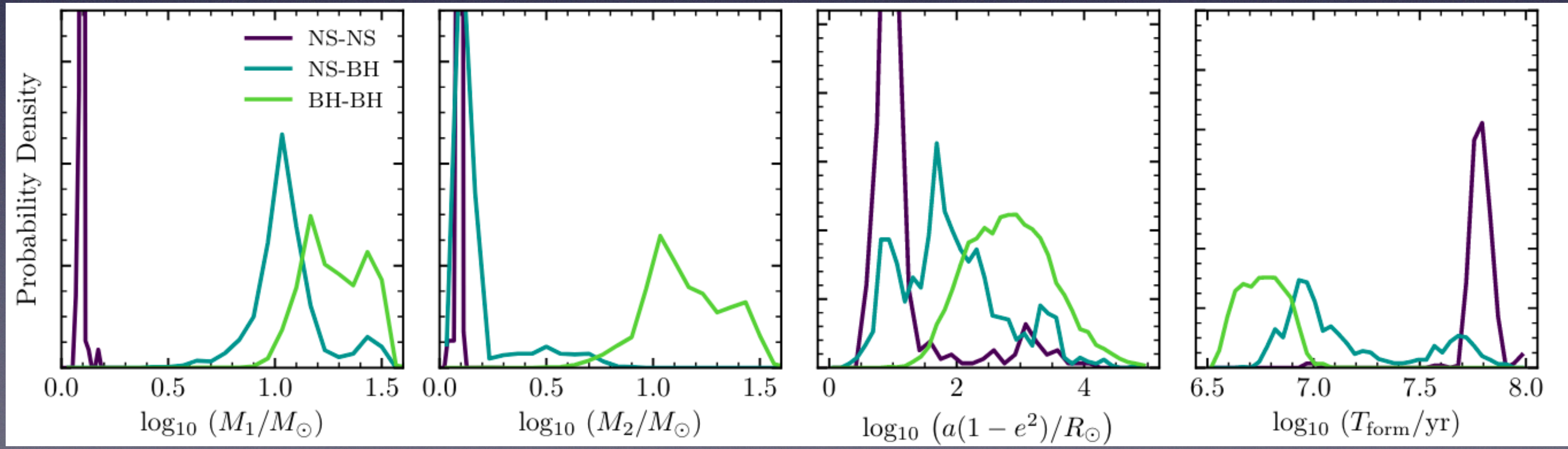
Binary Evolution Codes

- Cambridge STARS code (Eggleton 1971; Pols et al. 1995; Eldridge & Tout 2004)
- ev/STARS/TWIN (Pols et al.1995; Nelson & Eggleton 2001; Eggleton & Kiseleva-Eggleton 2002)
- BEC (Heger et al. 2000; Heger & Langer 2000)
- BINSTAR (Siess et al. 2013)
- MESA (Paxton et al. 2011, 2013, 2015, 2018, 2019)
- BPASS (Eldridge & Stanway 2016; Stanway et al. 2016; Eldridge et al. 2017; Stanway & Eldridge 2018)

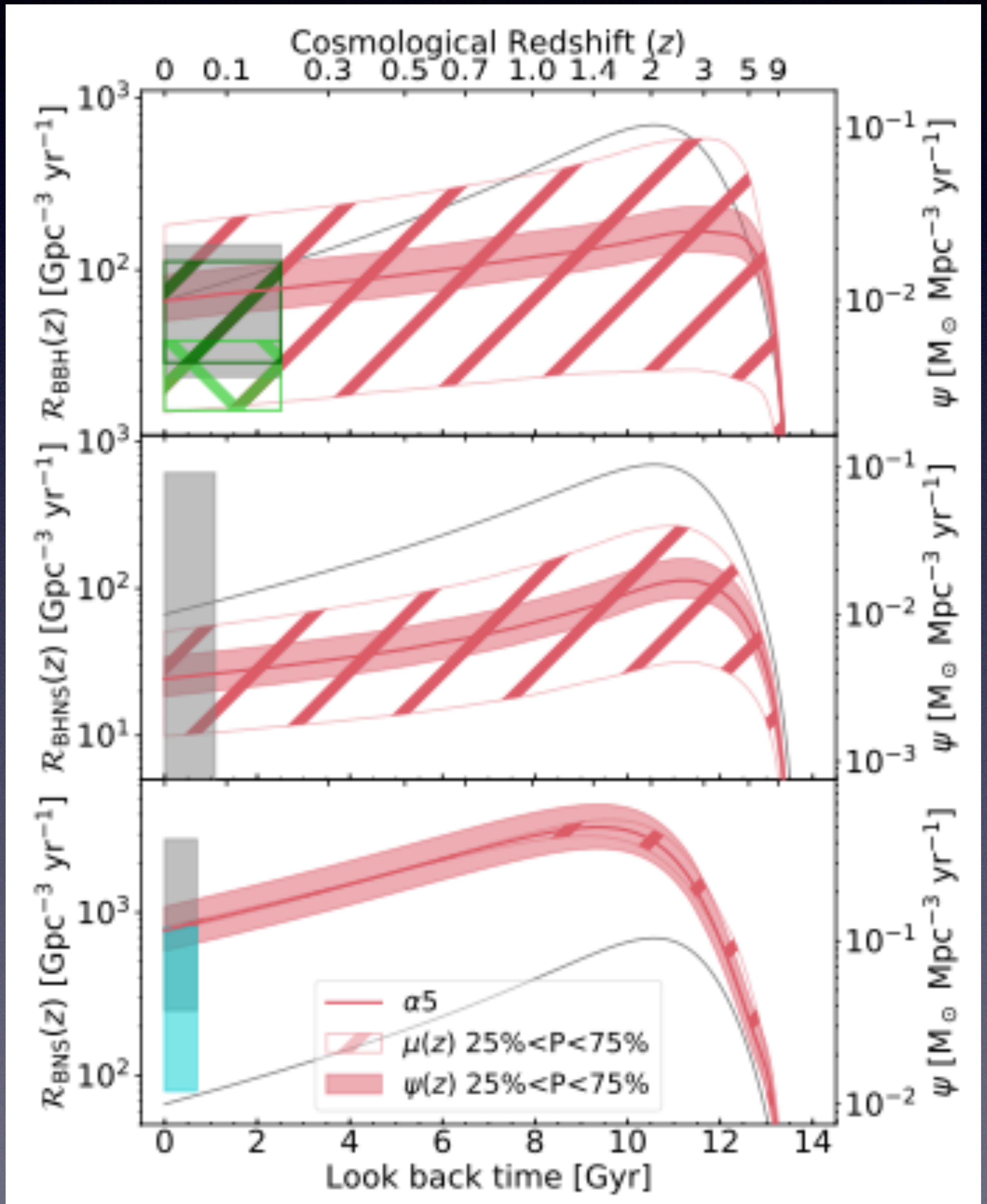
State of the art in massive binary simulations and inferences on progenitors

Predictions

- Statistics on merger rate density within the Hubble time.
- Statistics on total population of DCO.



DCO population produced by POSYDON. Credits Fragos et al. (2022)



Merger rate density of DCO. Credits Santoliquido et al. (2020)

Comparisons with GWTC-2

- Given N_{obs} LIGO/Virgo detections $\{x\} = \{x_1, \ldots, x_{N_{obs}}\}$
- Also given hyper-parameters θ, with prior p(θ), we have :

$$p(\theta \,|\, \{x\}, N_{obs}) \propto p(\{x\}, N_{obs} \,|\, \theta) p(\theta)$$

- Which can be rewritten as :

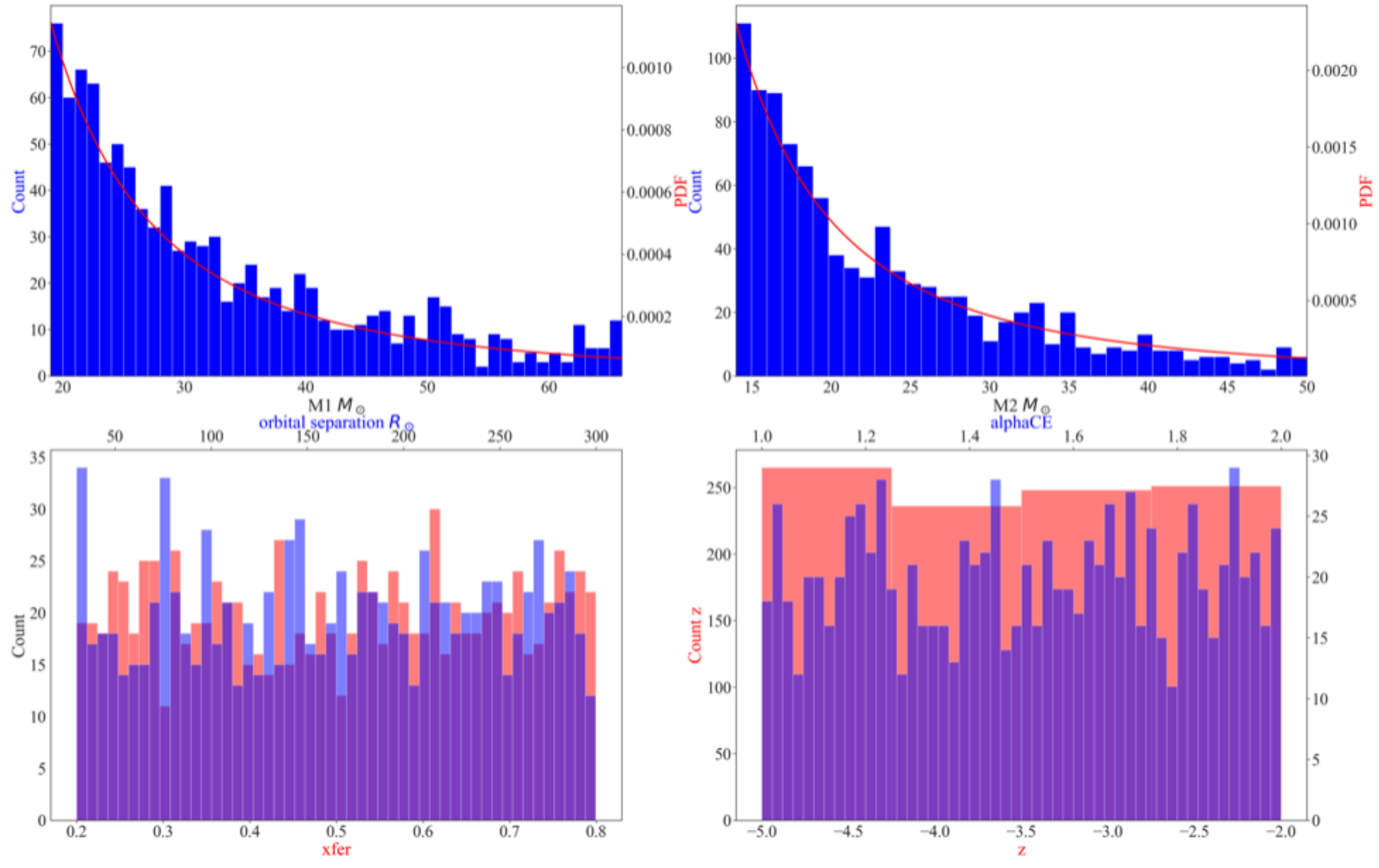
$$p(\theta \,|\, \{x\}, N_{obs}) \propto p(\theta) \prod_{i=1}^{N_{obs}} \frac{\int p(x_i \,|\, \theta, \alpha) p_{pop}(\alpha \,|\, \theta) d\alpha}{\int p_{det}(\alpha, \theta) p_{pop}(\alpha \,|\, \theta) d\alpha}$$

MCMC

Monte Carlo algorithm

Drawing a population

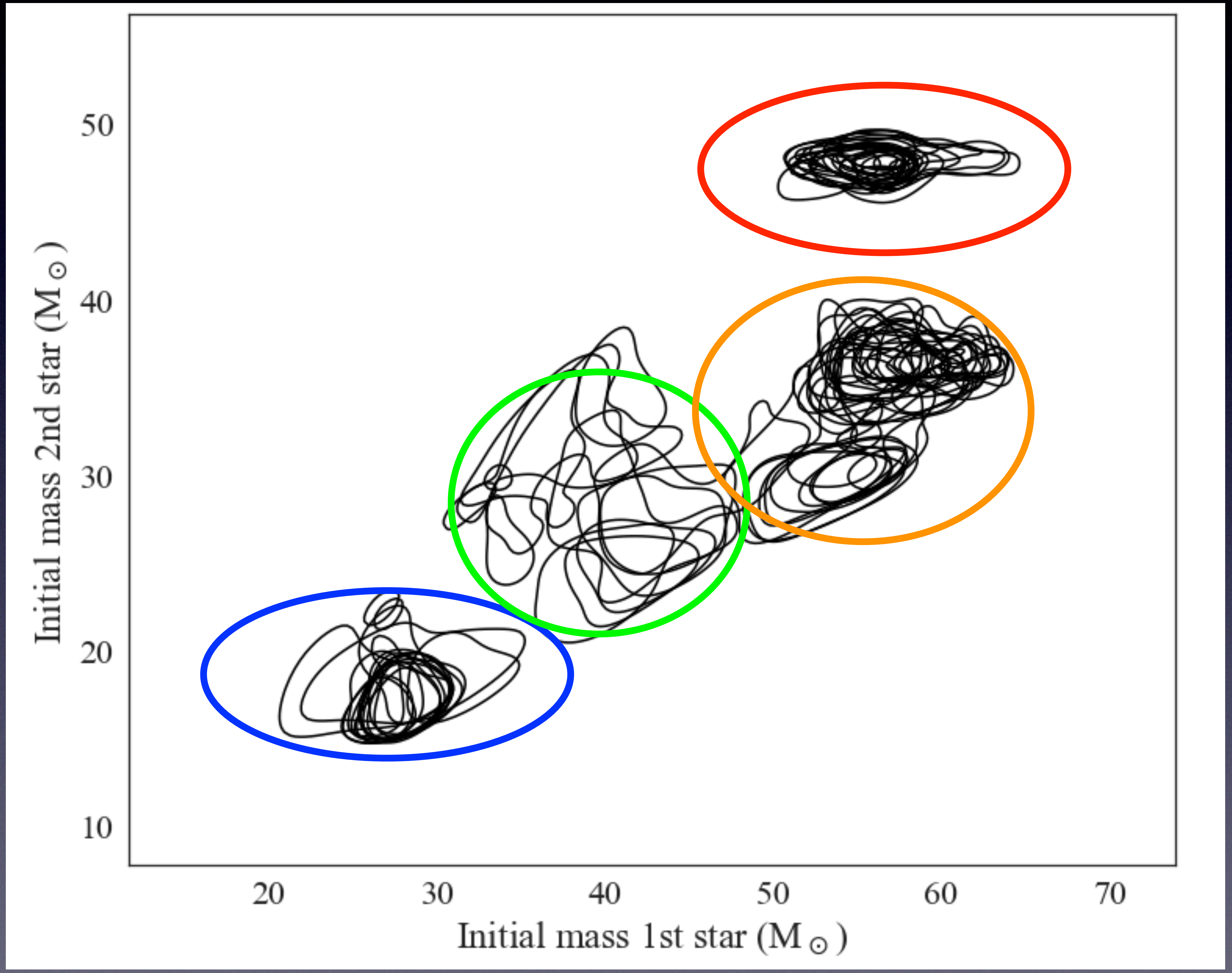
- Simulate a large population of binaries following priors.
- IMF : $p(m)dm \propto m^{-2.3}$ (Salpeter/Kroupa)
- Uniform distribution on metallicities, orbital separation, mass transfer and common enveloppe efficiencies.



Results : Mass distribution

- Several identified regions :
  - Low masses
  - Intermediate masses
  - High masses
  - High/intermediate masses

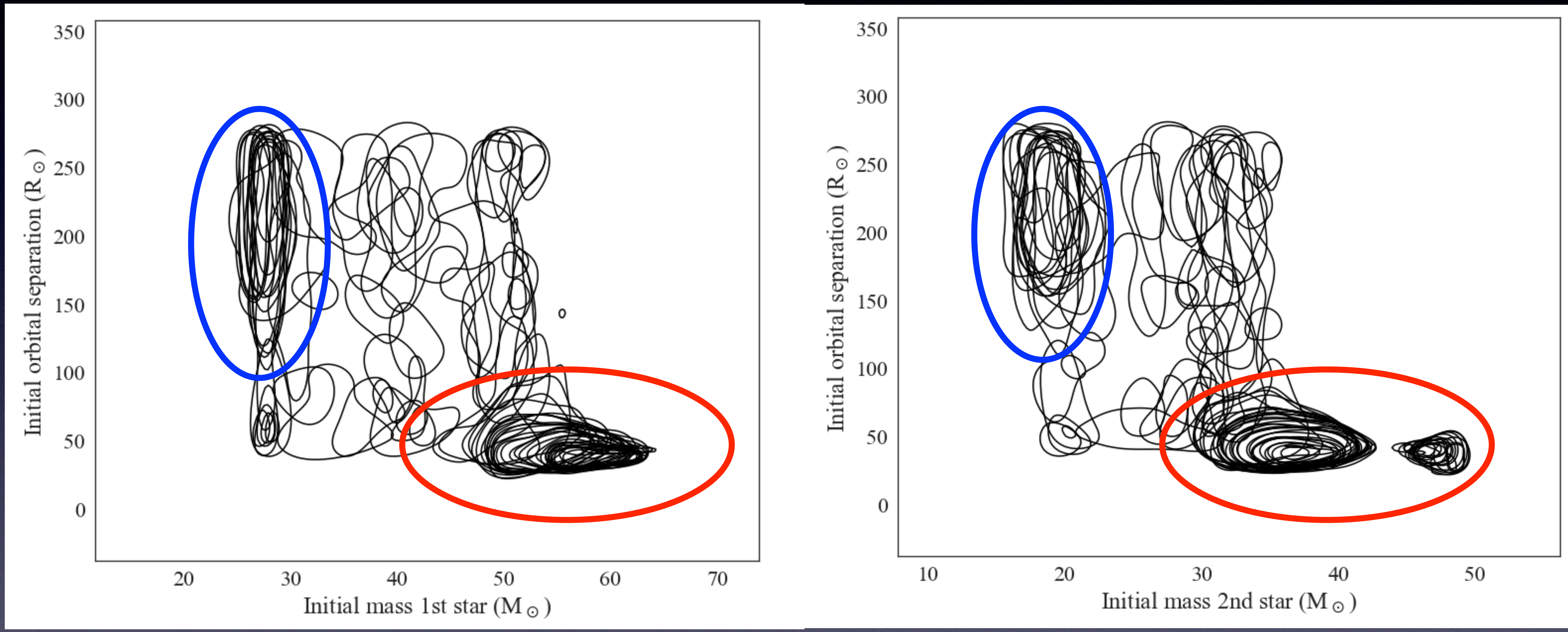
- Region not populated



Initial masses distribution of the events

Results : masses and orbital separation

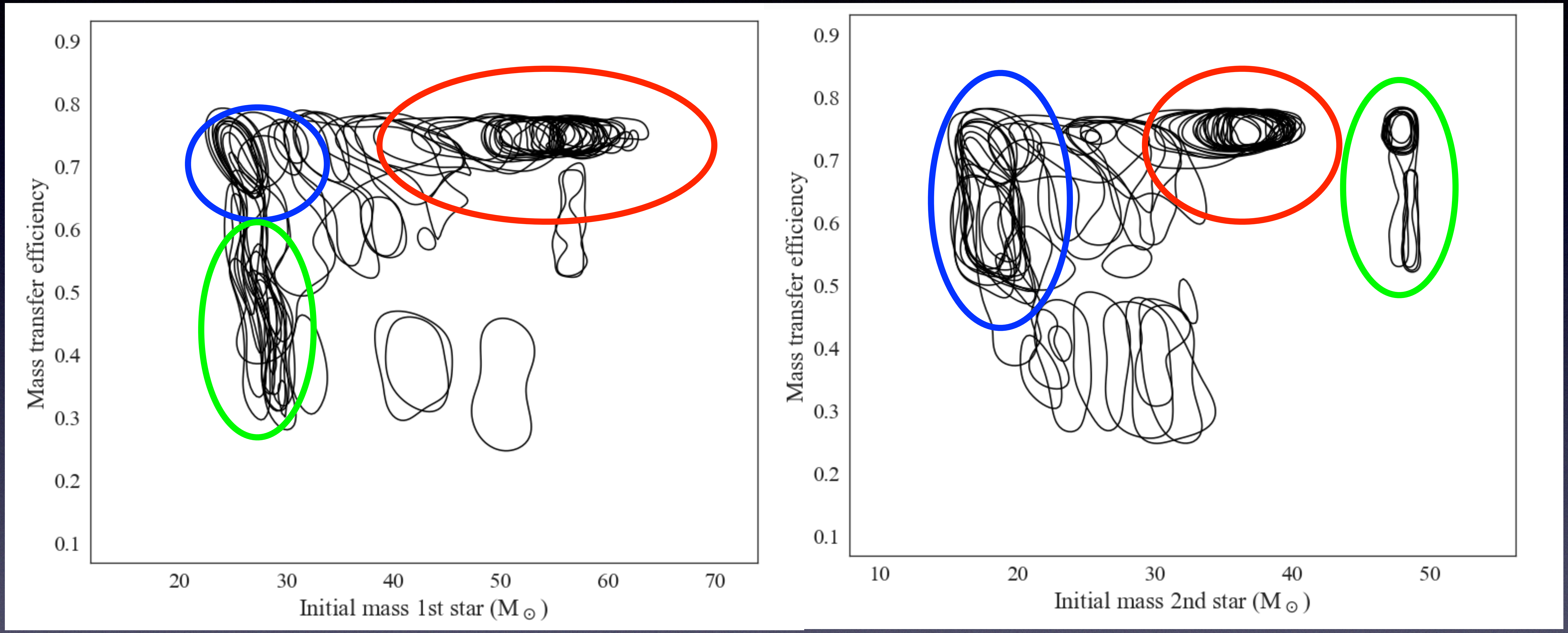
- Two regimes :
  -High masses/Low separation
  - Low masses/High separation



Results : Masses and mass transfer efficiency

- For m1 :
  - Low mass/High xfer
  - High mass/High xfer
  - Low mass/Low xfer

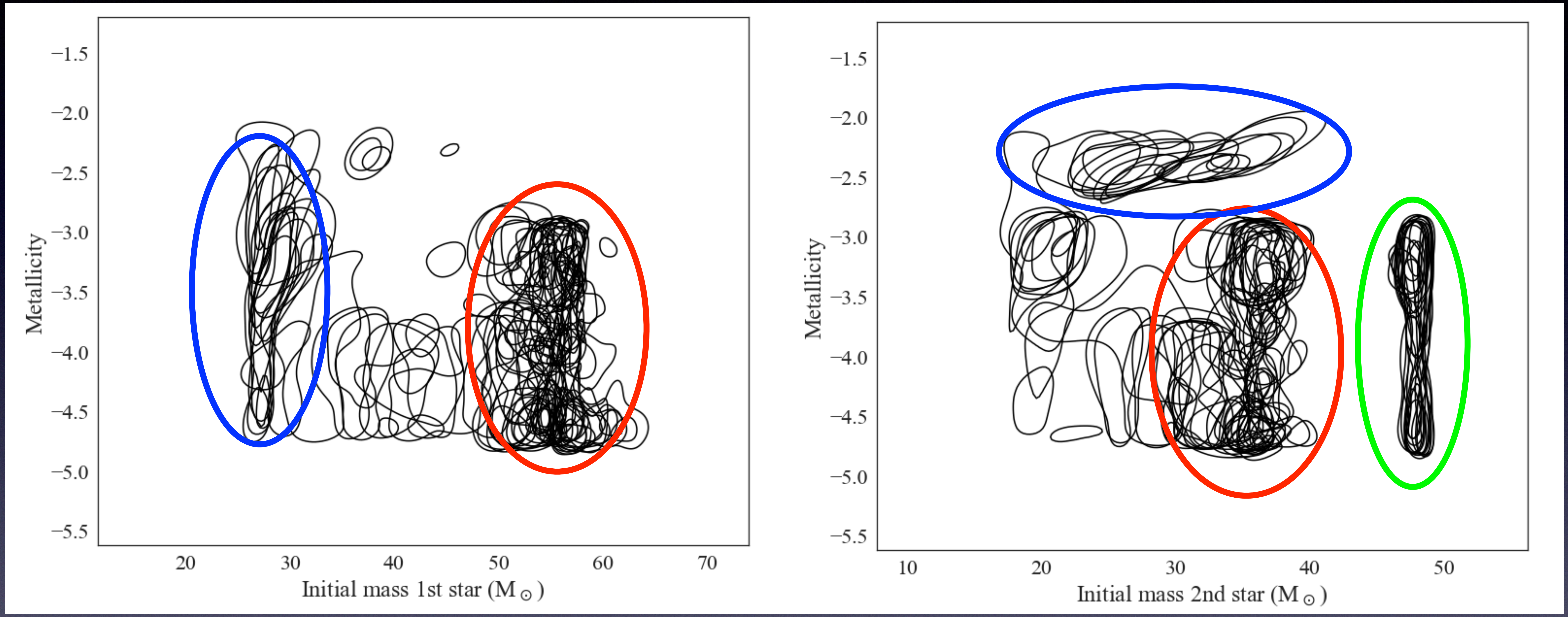
- For m2 :
  - Low mass/High xfer
  - High mass/High xfer
  - Very high masses/High xfer



WARNING : same color doesn't mean same sources

Results : Masses and metallicity

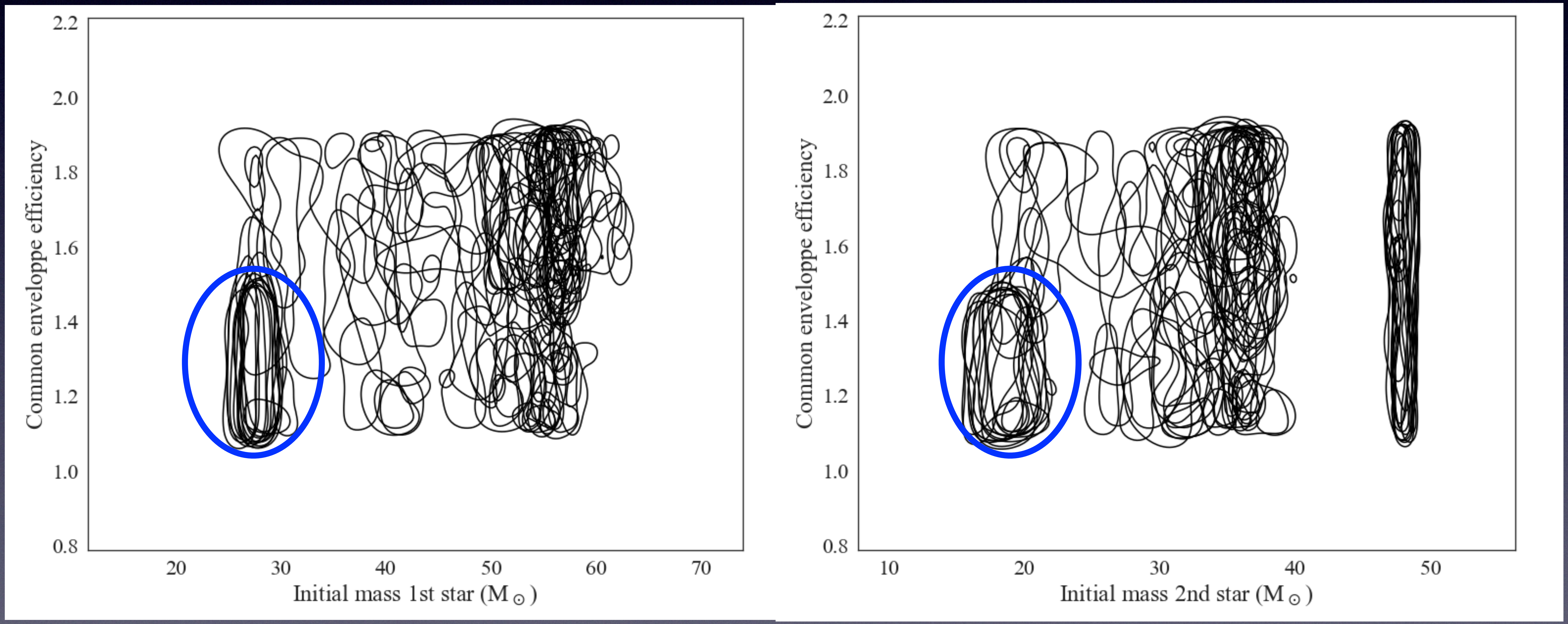
- For m1 :
  - High mass/Low z
  -Low mass/Higher z
- For m2 :
  - Low mass/High z
  - High mass/Low z
  - Very high masses/Low z



WARNING : same color doesn’t mean same sources

Results : Masses and Common Envelope efficiency

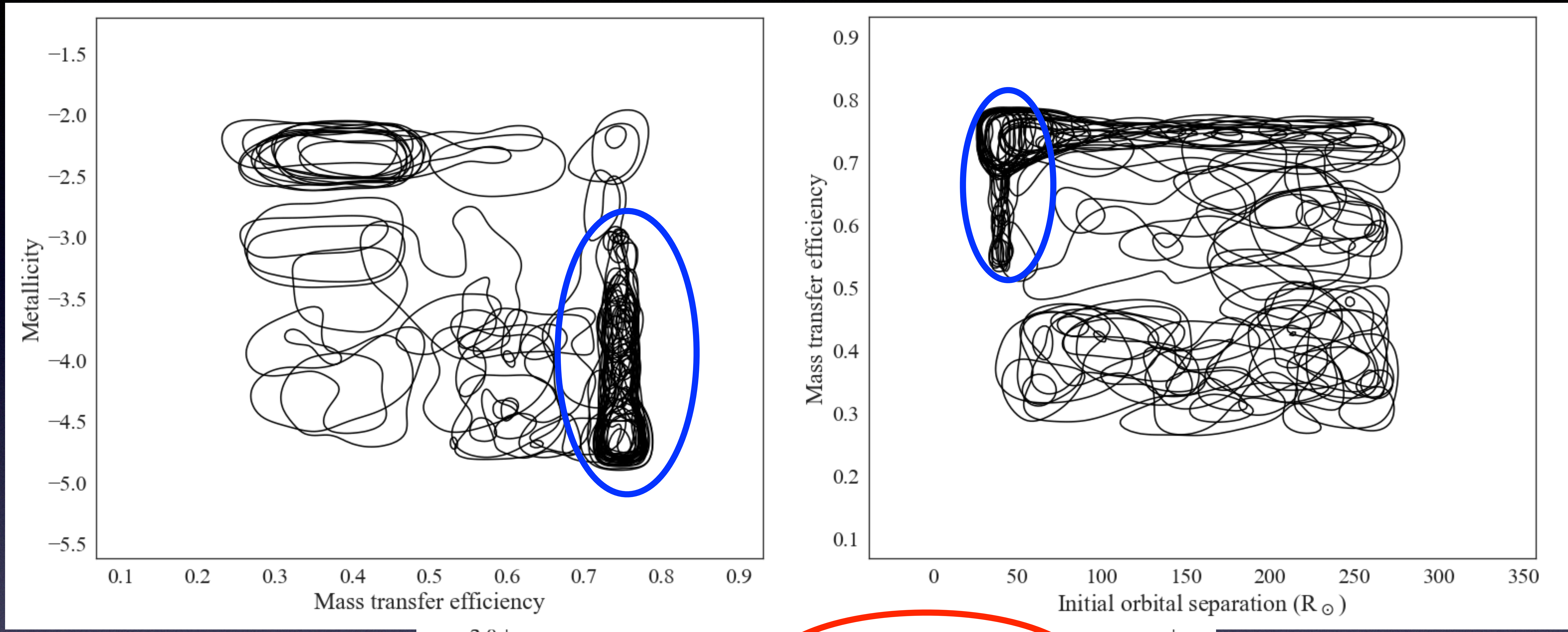
- One main behavior : Low mass/Low CE

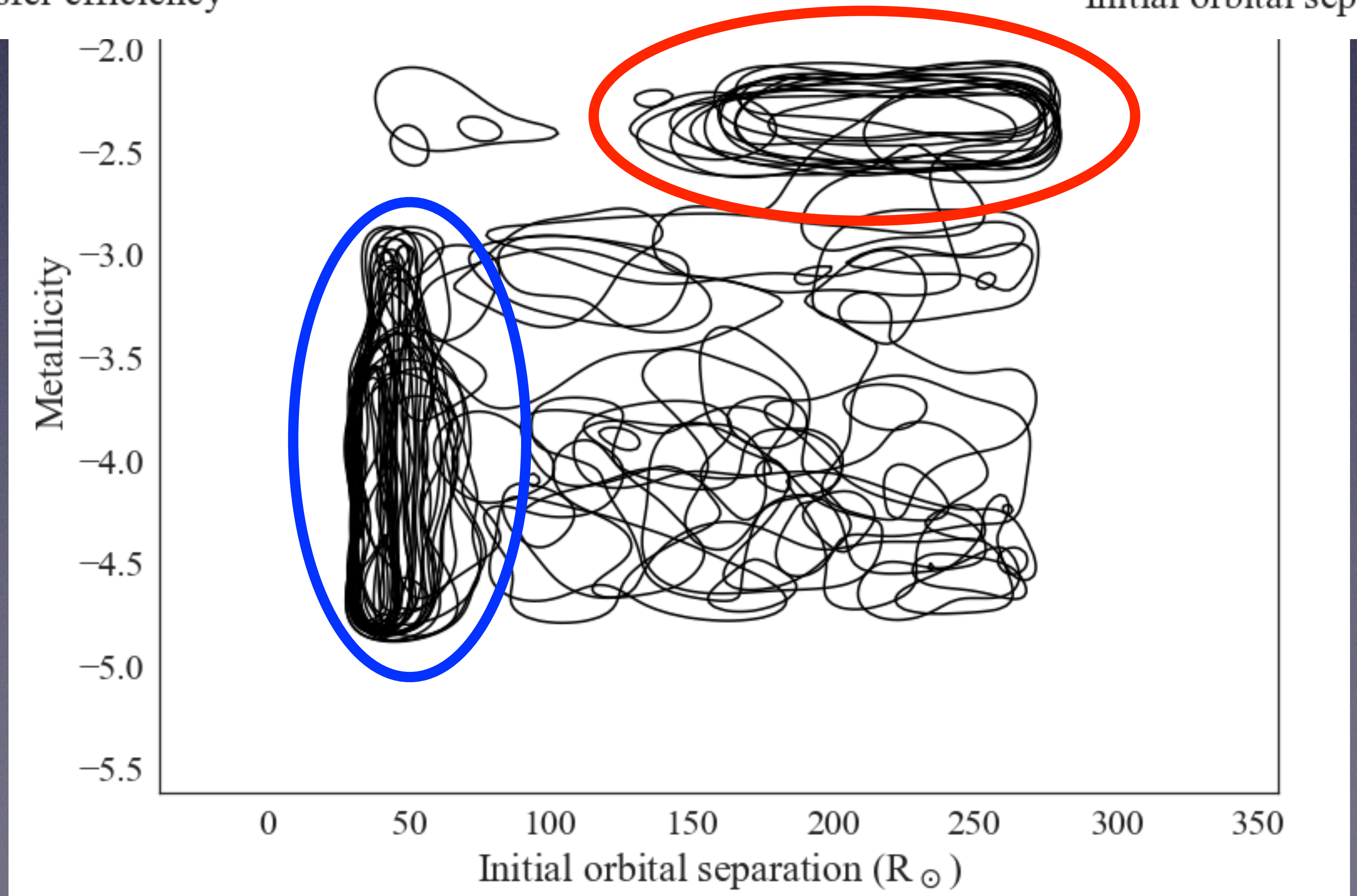


Results : Mass transfer, Metallicity and orbital separation

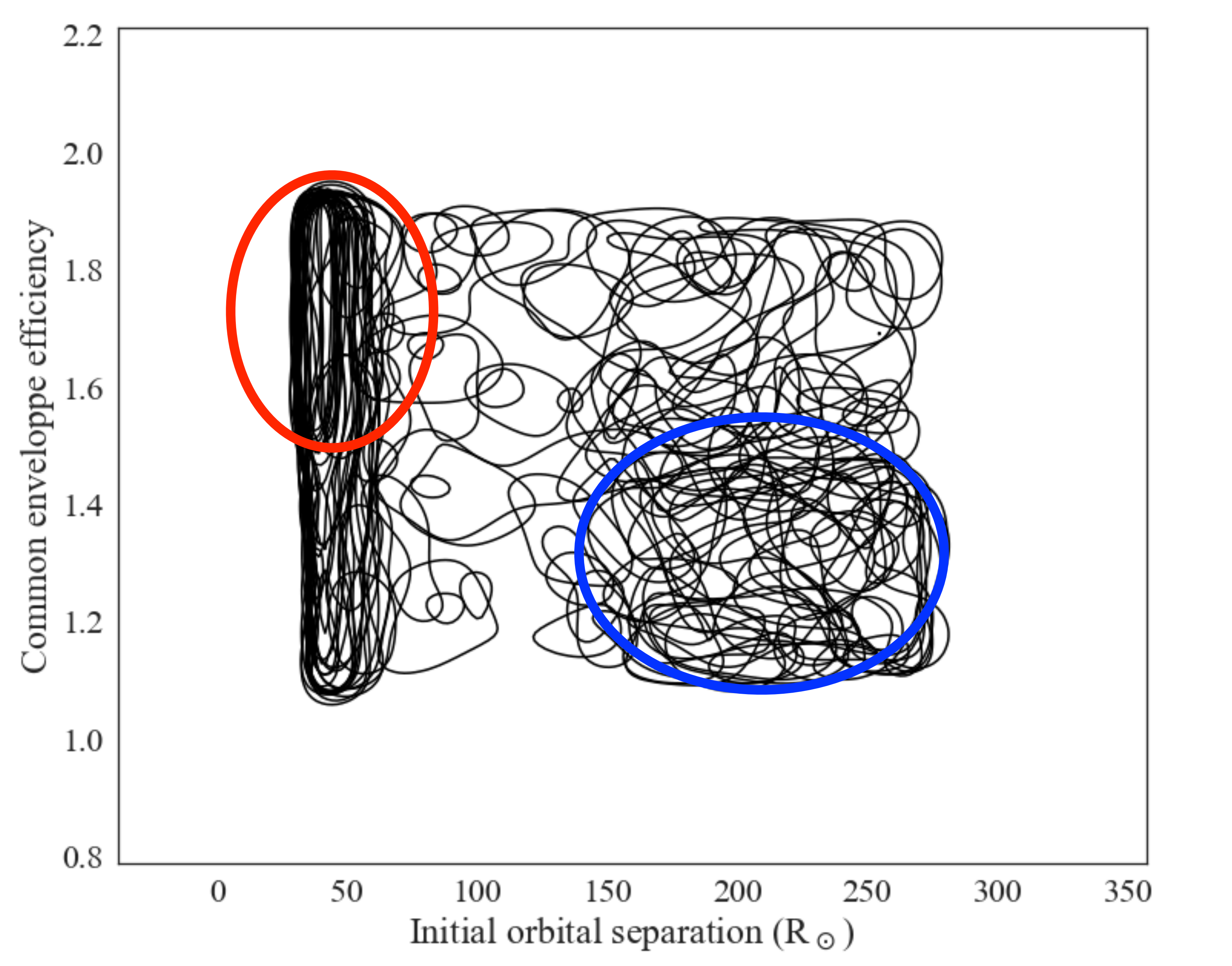
- High xfer/Low z
  Low separation/ High xfer
  Low separation/Low z

- High z/High separation

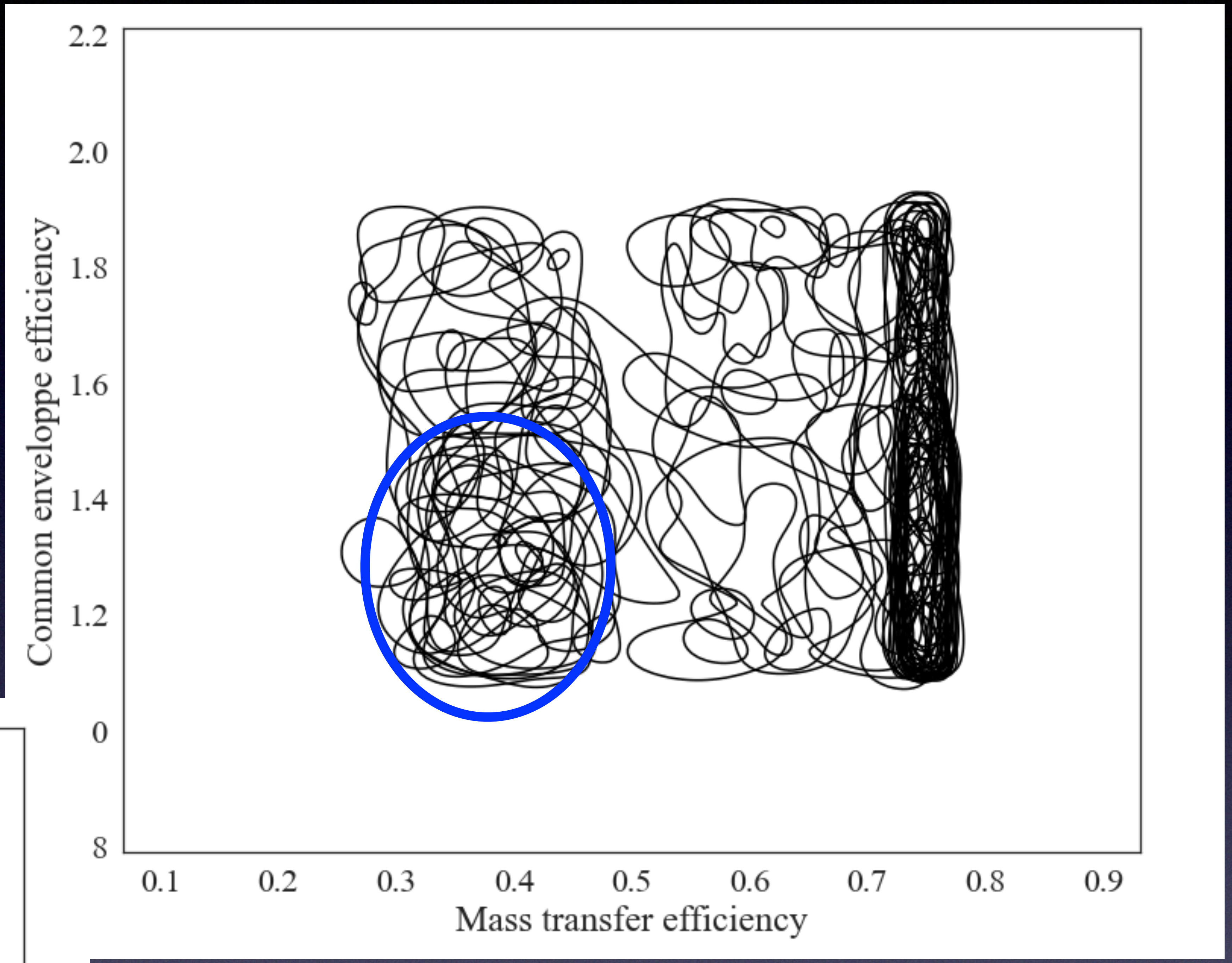


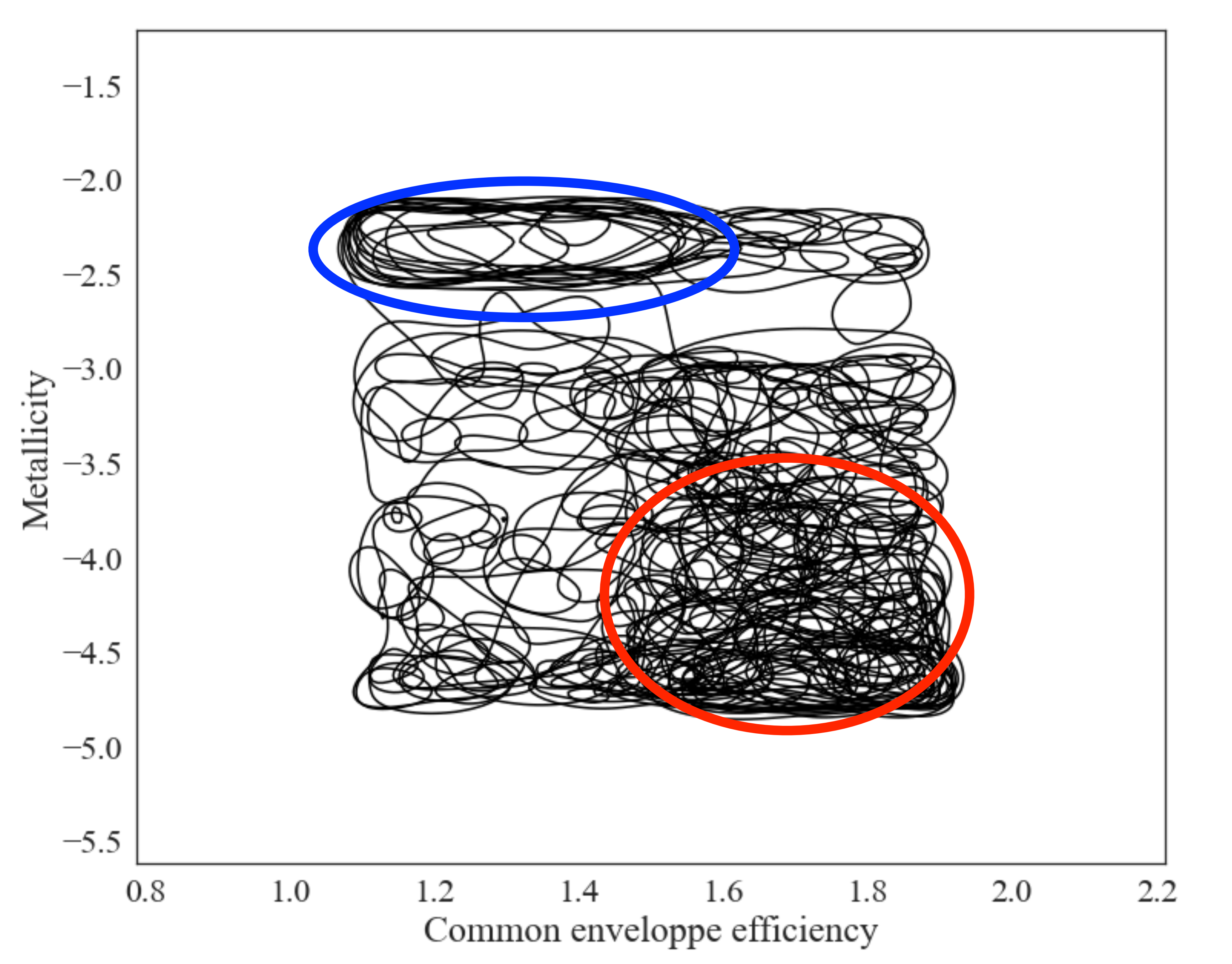


Results : Common enveloppe efficiency



- Low CE/High z
- High CE/Low z





- Low CE/High separation
- High CE/Low separation

- Low CE/Low xfer