Chempy: Finding Objective Scores for Nucleosynthetic Yield Tables

Oliver Philcox (IoA), Jan Rybizki (MPIA)

Milky Way Group Meeting

August 9, 2017

Outline

Motivation

- All galactic chemical evolution models (GCEs) depend on a set of nucleosynthetic yield tables.
- These give the elemental yields for a particular process e.g.
  - Type Ia Supernovae (Nomoto et al. 2013, Chieffi et al. 2004)
  - Core Collapse Supernovae (CC-SN) (Seitenzahl et al. 2013, Thielemann et al. 2003)
  - AGB Stars (Karakas 2010 & 2016, Ventura et al. 2013)
- Create a metric to determine how well these reproduce solar abundances

Chempy

Background

- Implemented GCE model is *Chempy* (Rybizki et al. 2017)
  - "A parameterized open one-zone model within a Bayesian Framework"
  - **Input:** 3 global Simple Stellar Population (SSP) parameters & 3 local ISM parameters
  - **Output:** ISM elemental abundances at time of solar birth
- Observational data is proto-solar abundances (Turcotte & Wimmer-Schweingruber 2002)
- *Chempy* calculates the best-fit parameters using MCMC sampling (Foreman-Mackey et al. 2013) to maximize the likelihood function

Chempy

Free parameters

Table 1. Free Chempy parameters, $\vec{\theta}$, and their priors with assumed Gaussian error model.

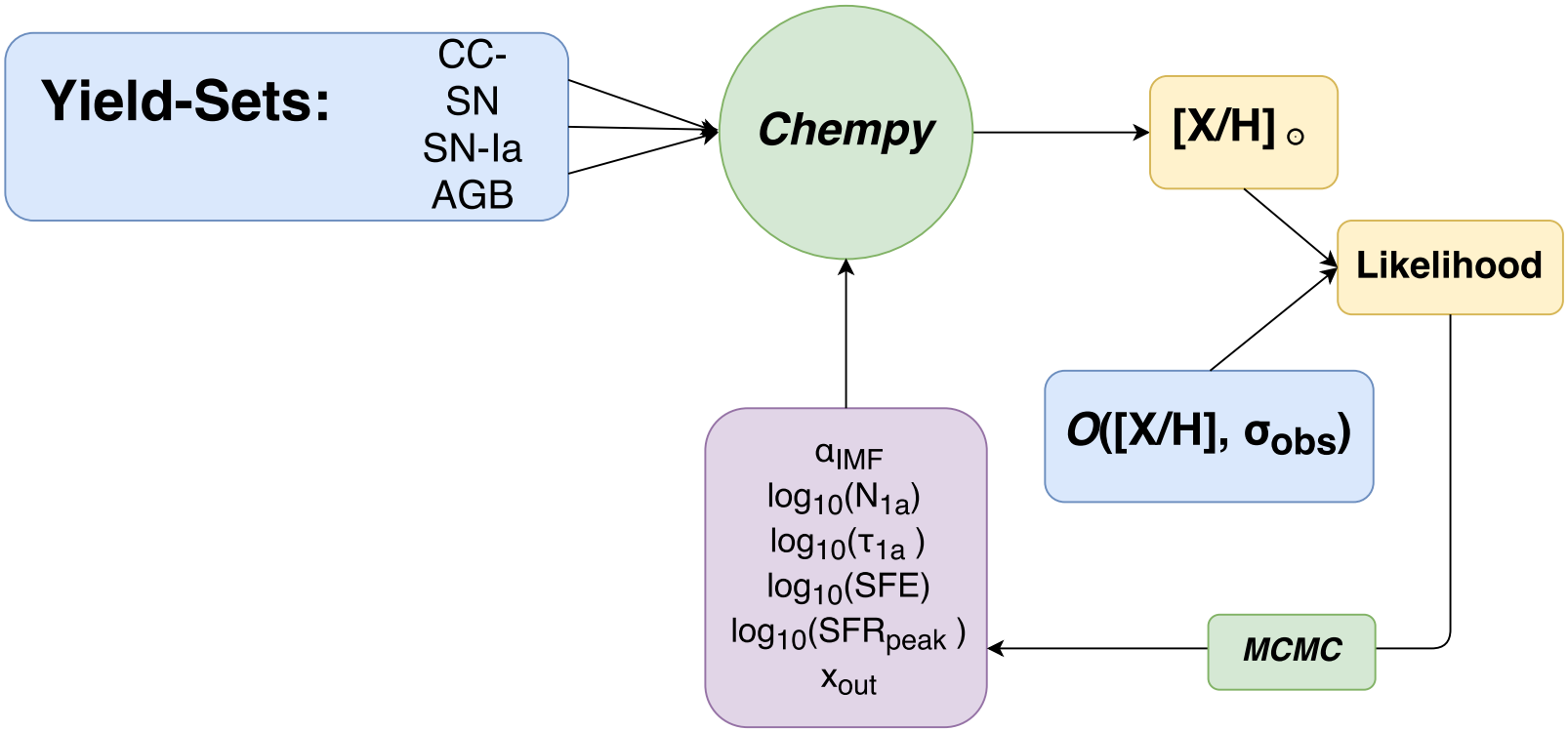
$\vec{\theta}$	description	$\bar{\theta}_{\rm prior} \pm \sigma_{\rm prior}$	limits	approximated prior based upon
	global stellar (SSP) parameters			
$\alpha_{\rm IMF}$	high-mass slope of the Chabrier (2003, tab.1) IMF	-2.3 ± 0.3	$[-4, -1]$	Chabrier (2003, tab.1)
$\log_{10}(\rm N_{Ia})$	number of SN Ia exploding per $\rm M_\odot$ over 15 Gyr	-2.75 ± 0.3	$[-5, -1]$	Maoz & Mannucci (2012, tab. 1)
$\log_{10}(\tau_{\rm Ia})$	SN Ia delay time in Gyr until first occurrence	-0.8 ± 0.3	$[-3, 1]$	Maoz et al. (2012)
	local ISM parameters			
$\log_{10}(\rm SFE)$	star formation efficiency governing the infall and ISM	-0.3 ± 0.3	$[-3, 2]$	?
$\log_{10}(\rm SFR_{peak})$	peak of SFR in Gyr (scale of γ-distribution with k=2)	0.55 ± 0.1	$[-1, 1]$	van Dokkum et al. (2013, fig 4b)
$\rm x_{out}$	fraction of stellar feedback outflowing to the corona	0.5 ± 0.1	$[0, 1]$	guessed borad prior

(Philcox & Rybizki, in prep.)

Chempy

Calculation Schematic

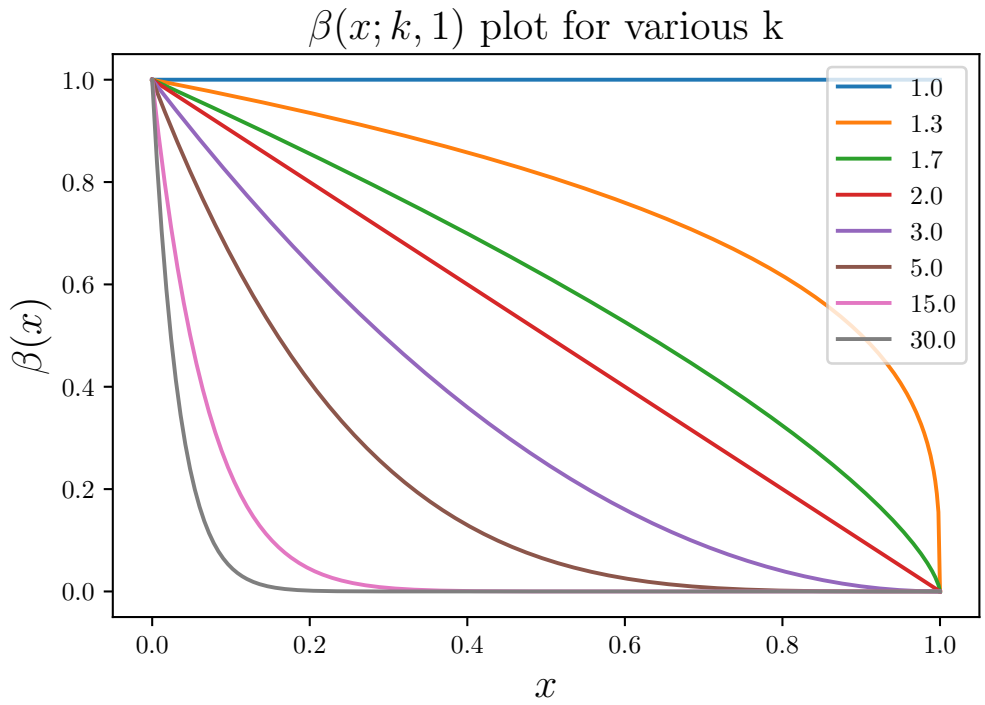
MCMC Parameter Optimization:



Chempy

Error Function

- For each element, we marginalize over the model error, $\sigma_{\rm model}$, to account for unmodelled effects.
- Error function is a $\beta(x; k, 1)$ function, for abundance x.
- k-parameter controls the error strength (default $= 3$).



Chempy Posteriors

Optimal parameter set

Posteriors were calculated for two choices of yield sets (hyperparameters):

- **CC-SN**: Nomoto et al. (2013)
- **SN-Ia**: Seitenzahl et al. (2013)
- **AGB**: Karakas (2010) or Karakas (2016)

Table 2. Inferred parameters (16, 50, 84 percentiles of $\theta_{\rm posterior}$, for different observational subsets ($\mathcal{O}_{\rm s}$) and different yield sets.

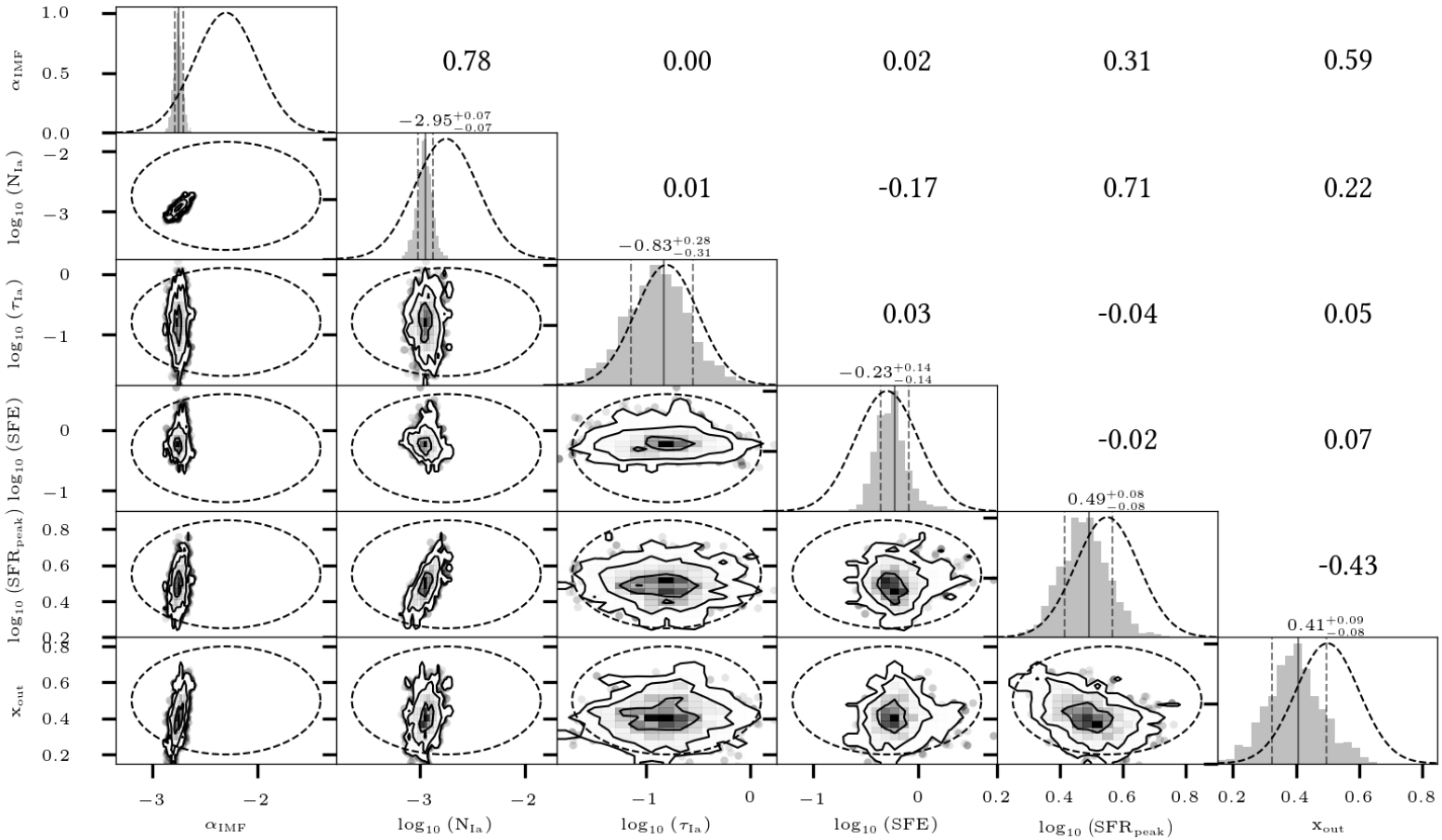
observational set	posterior	$\alpha_{\rm IMF}$		$\log_{10}(\rm N_{1a})$		$\log_{10}(\tau_{\rm 1a})$		$\log_{10}(\rm SFE)$		$\log_{10}(\rm SFR_{peak})$		$\rm x_{out}$	
$\mathcal{O}_{\rm s}$	$\mathbb{P}_{\rm max}$	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
$\binom{\rm Karakas10}{\rm With\ error}$	1.43	-2.58	$^{+0.11}_{-0.12}$	-2.82	±0.15	-0.86	$^{+0.31}_{-0.27}$	-0.31	$^{+0.31}_{-0.24}$	0.55	±0.10	0.51	±0.11
$\binom{\rm Karakas10}{\rm No\ error}$	-83.41	-2.80	±0.05	-3.06	$^{+0.07}_{-0.11}$	-0.83	$^{+0.32}_{-0.29}$	-0.26	±0.11	0.43	$^{+0.06}_{-0.07}$	0.36	±0.09
$\binom{\rm Karakas16}{\rm With\ error}$	1.60	-2.58	±0.12	-2.80	±0.14	-0.79	$^{+0.28}_{-0.29}$	-0.33	$^{+0.29}_{-0.24}$	0.54	$^{+0.10}_{-0.11}$	0.49	±0.10
$\binom{\rm Karakas16}{\rm No\ error}$	-84.01	-2.75	±0.04	-2.95	±0.07	-0.83	$^{+0.28}_{-0.31}$	-0.23	±0.14	0.49	±0.08	0.41	$^{+0.09}_{-0.08}$

(Philcox & Rybizki, in prep.)

Chempy

Corner Plot

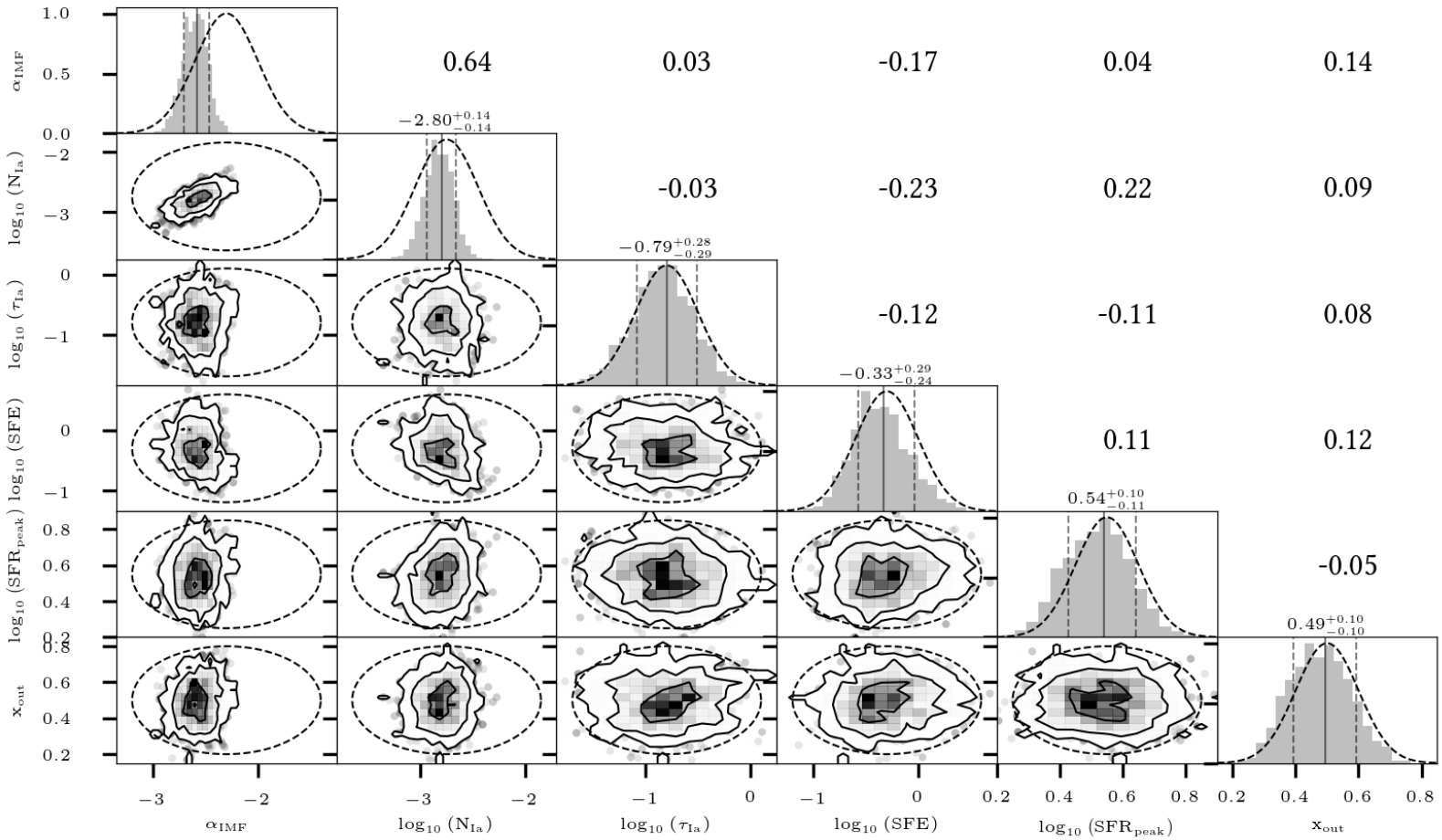
Output for Karakas (2016) AGB yields



Chempy

Corner Plot + Errors

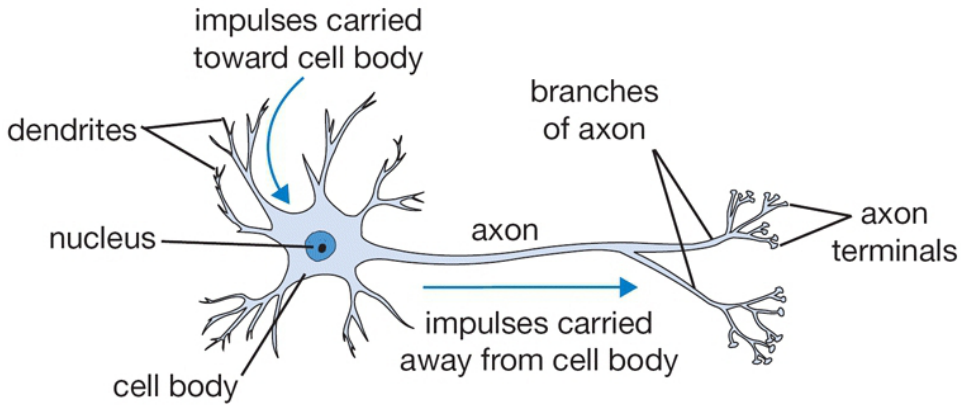
Output for Karakas (2016) AGB yields with β error model



Neural Networks

Motivation

- The *Chempy* code takes $\approx$ 3s to run per set of parameters queried
- This is too slow for efficient MCMC sampling and future parameter-space integration:
  - Train a neural network to predict the output abundances from input parameters
  - This runs in 7 ms here

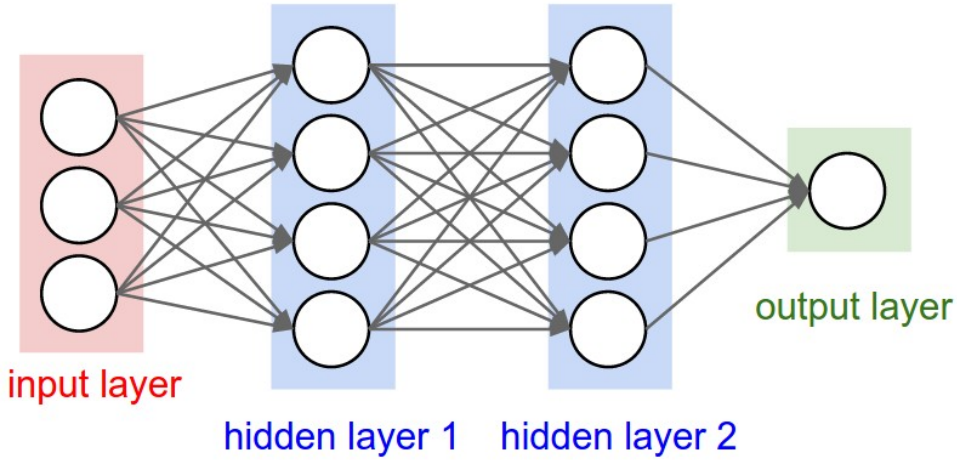


Karparthy et al. (2017)

Neural Networks

Network Structure

Network schematic:



Karparthy et al. (2017)

Our implementation:

- 6 inputs (parameter vector θ)
- 22 outputs ([X/Fe] abundances)
- 1 hidden layer with 30 neurons

Neural Networks

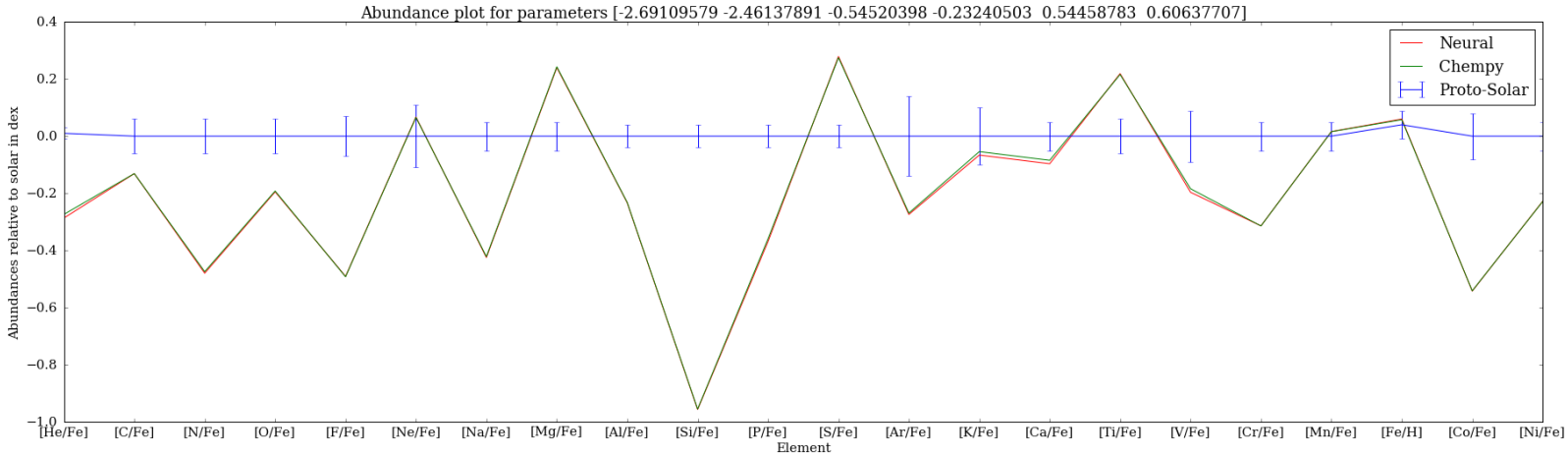
Network Structure

- Network created using *PyTorch* (`pytorch.org`)
- Settings used:
  - Adam sampler (Kingma & Ba 2014)
  - L1 loss function
  - tanh activator function
  - *PyTorch*'s `ReduceLROnPlateau` function
  - 5000 training epochs

Neural Networks

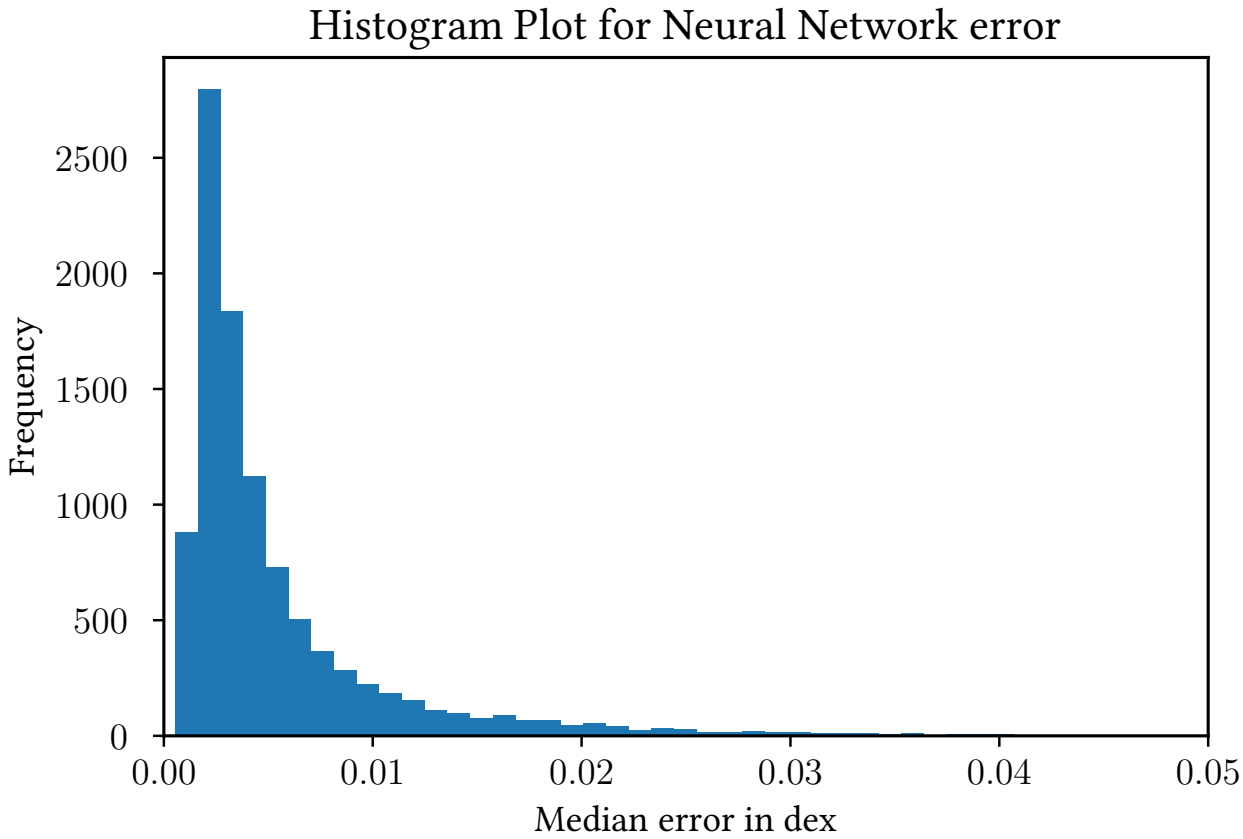
Training the Network

1. Create network architecture
2. Load 5^6 element training data-set, with parameters evenly distributed in Gaussian prior space
3. Train neural network with data-set
4. Optimize learning rate using random 10,000 element verification data-set (0.003 here)
5. Test settings with a further random 10,000 element data-set



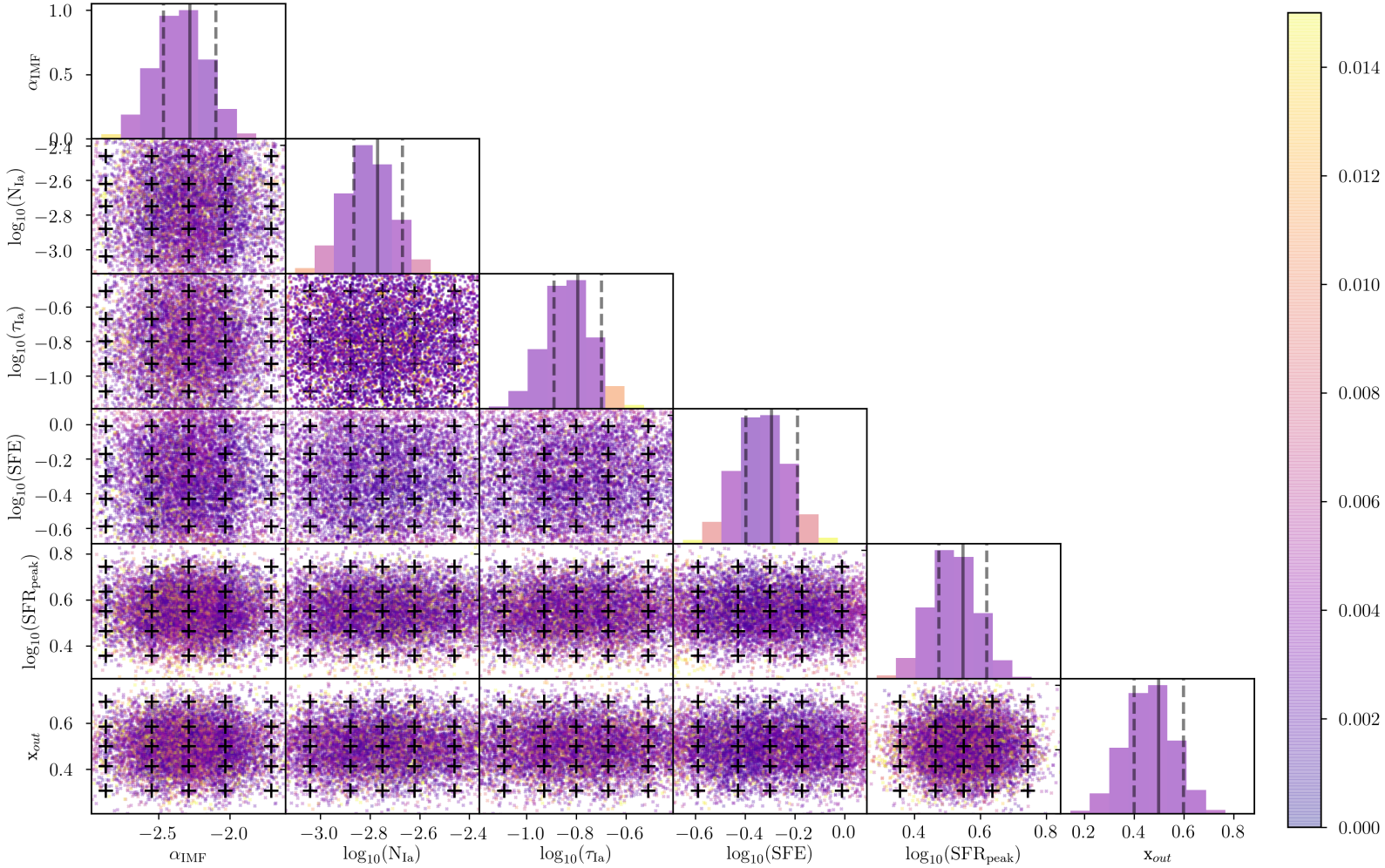
Testing the Network

Error Histogram



Testing the Network

Median Element Error Corner Plot



How can we create an objective scoring for yield tables?

- Use Bayes' factor to compare yield sets

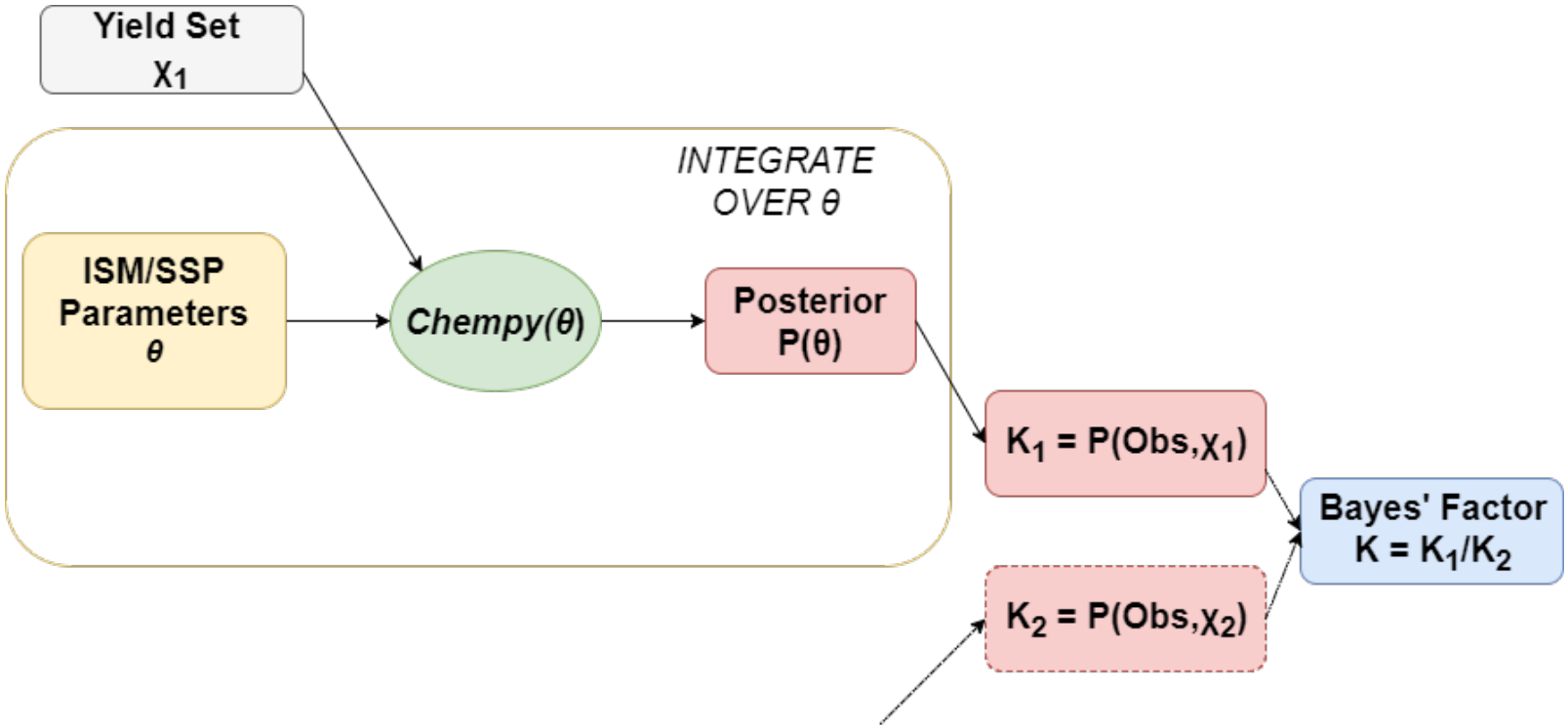
$$K = \frac{K_1}{K_2};$$

$$K_k = \mathrm{P}(\mathcal{O}|\chi_k) = \int_{\vec{\theta}} \mathrm{P}(\mathcal{O}|\vec{\theta}, \chi_k) p(\vec{\theta}) \mathrm{d}\vec{\theta}$$

  for observations $\mathcal{O}$ and yield sets χ, with prior $p(\vec{\theta})$ on the parameters.
- This is just the integrated posterior in 6 dimensions.
- If $K \gg 1$ then χ_1 better represents solar abundances than χ_2.
- Using the neural networks, we can calculate this integral using Monte Carlo methods e.g. `mcint` (Snowsill, 2011)

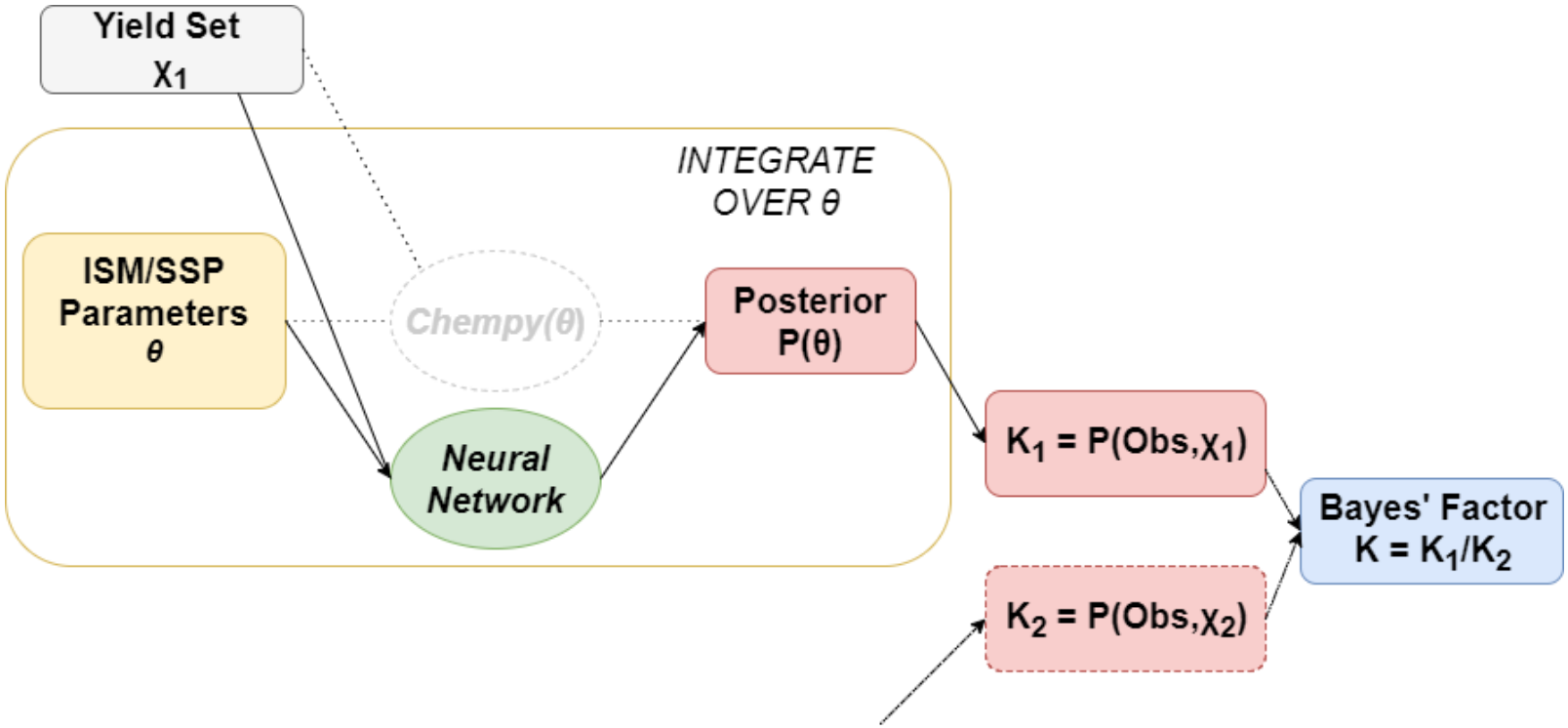
Objective Scoring Outline

With *Chempy*



Objective Scoring Outline

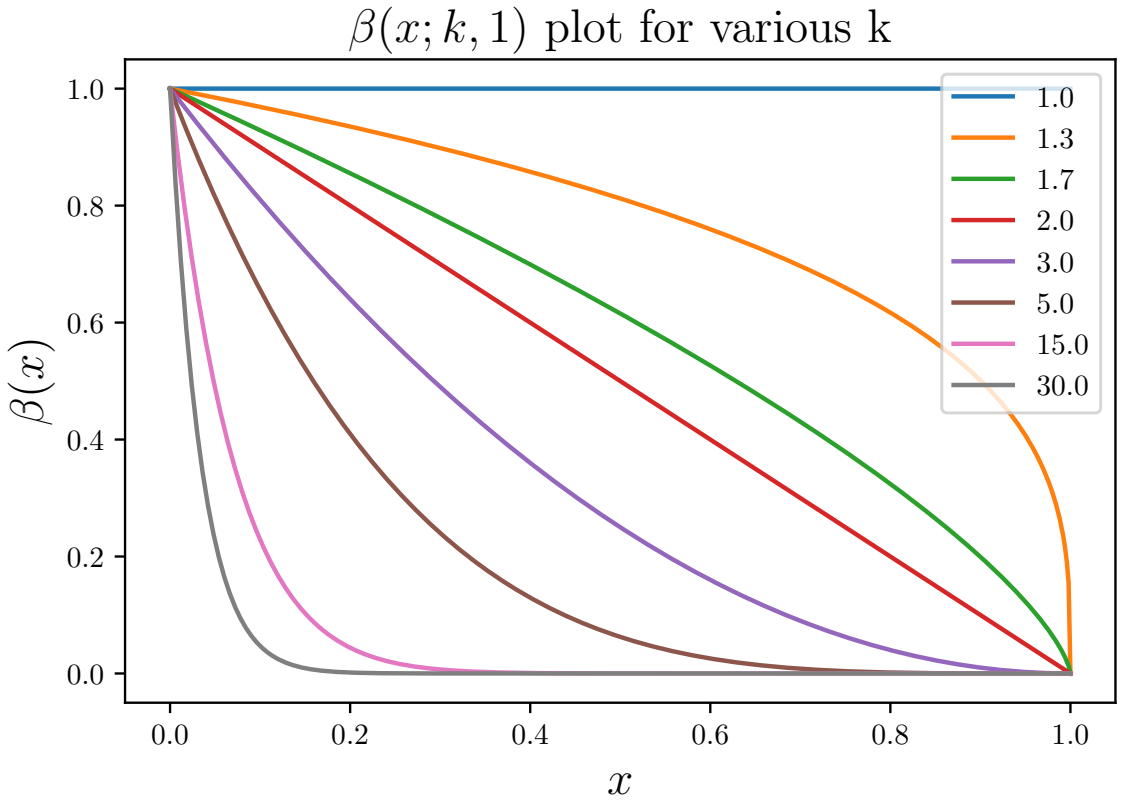
With Neural Network



Objective Scoring

Elements & Errors

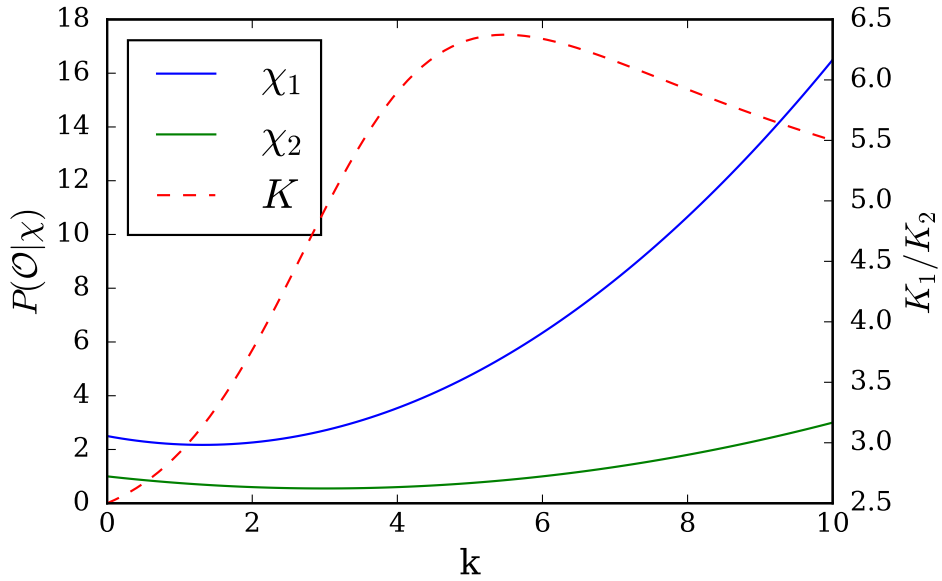
- Which elements should we fit for?
  - Probably all elements else we introduce additional bias.
- The model error in each element of *Chempy* is not known.
- Currently we marginalize over the error during each posterior calculation, using a β model.



Objective Scoring

Elements & Errors

- One option:
  - Parameterize the error using the $\beta(x, k, 1)$ k-parameter
  - Calculate $K(k)$ across k-space
  - As $k \to \infty$, the error tends to 0
  - As $k \to 1$ the error tends to 1 dex
- Any other suggestions?



Further Work

- Implement objective scoring function into *Chempy*
  - Widen the training data-set to span 3σ per parameter
  - Retest neural network
  - Integrate over parameter space using Monte Carlo methods
  - Calculate objective score
- Test approach on different sets of yield tables