The data-driven approach to measuring stellar parameters

Tuesday, June 4, 2019 — 2.15-3.30

Adam Wheeler (Columbia) a.wheeler@columbia.edu

Should have been: Melissa Ness (Flatiron/Columbia, NYC)
mkness@gmail.com

Outline

Outline

- Motivate data-driven approach to spectroscopy
- The Cannon - how it works
- Applications
  - Precision abundances
  - Low-res spectra
  - Stellar ages

Data Analysis principles

Data Analysis principles

- respect and understand your data (plot your spectra and look at it - a lot!)

Data Analysis principles

- respect and understand your data (plot your spectra and look at it - a lot!)
  - explore systematics in your data (understand the instrument!)

Data Analysis principles

- respect and understand your data (plot your spectra and look at it - a lot!)
  - explore systematics in your data (understand the instrument!)
  - examine and understand correlations between measurements,

Data Analysis principles

- respect and understand your data (plot your spectra and look at it - a lot!)
  - explore systematics in your data (understand the instrument!)
  - examine and understand correlations between measurements,
  - understand your errors

Data Analysis principles

- respect and understand your data (plot your spectra and look at it - a lot!)
  - explore systematics in your data (understand the instrument!)
  - examine and understand correlations between measurements,
  - understand your errors
  - making diagnostic plots is critical - question everything

Data Analysis principles

- respect and understand your data (plot your spectra and look at it - a lot!)
  - explore systematics in your data (understand the instrument!)
  - examine and understand correlations between measurements,
  - understand your errors
  - making diagnostic plots is critical - question everything

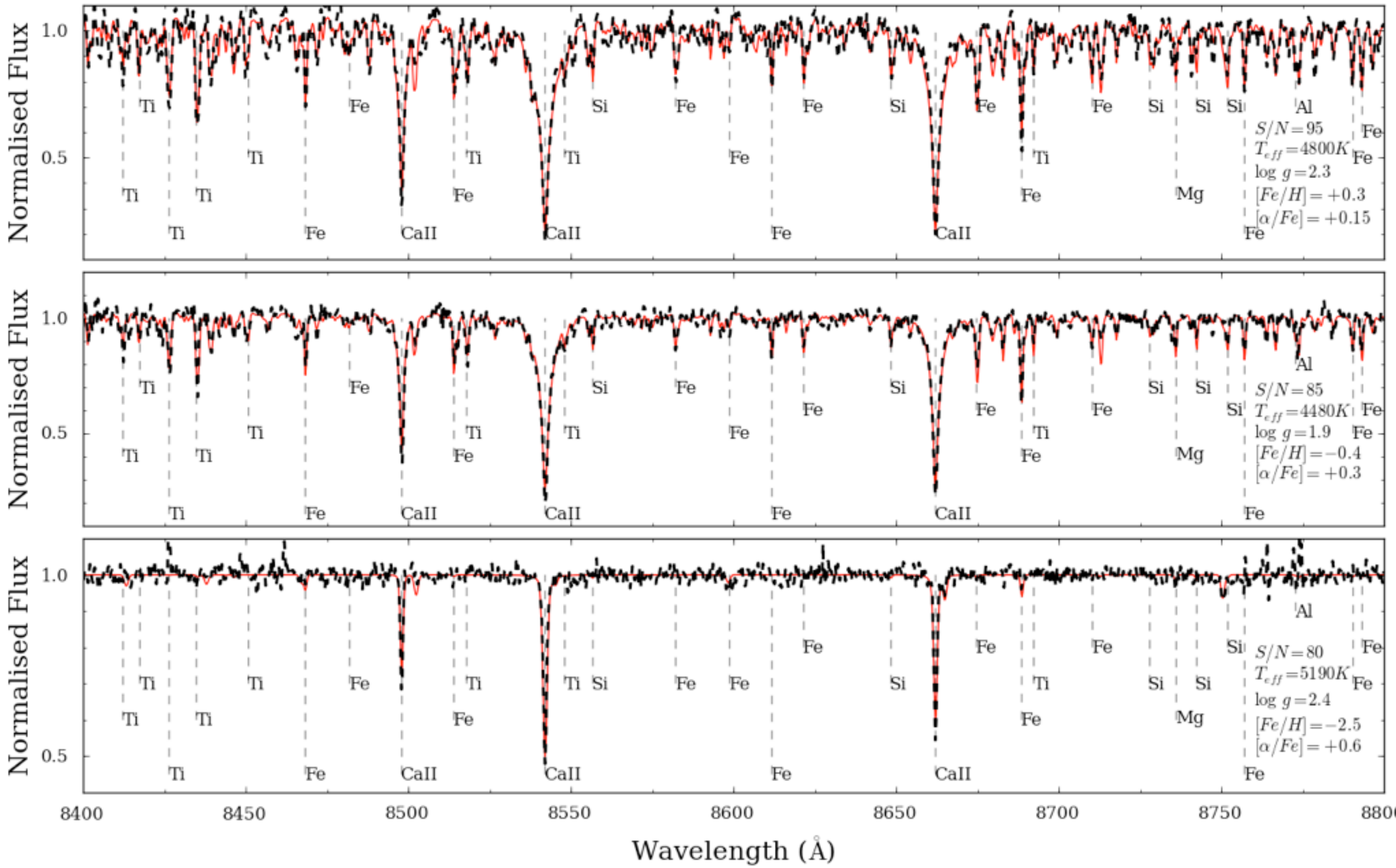
- start with simple, interpretable approaches and build in complexity as needed

Measuring stellar parameters with physics

1. Take model atmospheres from Castelli/Kurucz model atmospheres
2. Add a linelist from atomic and molecular library
3. MOOG (Sneden 1983) to create library of synthetic spectra.
4. Program to determine best fitting spectra in Teff, logg, [Fe/H], [α/Fe]
5. Calibrate: modify log gf values the Sun/Arcturus, compare results to literature and “benchmarks”, fit out trends

Stellar parameters measured

template
model



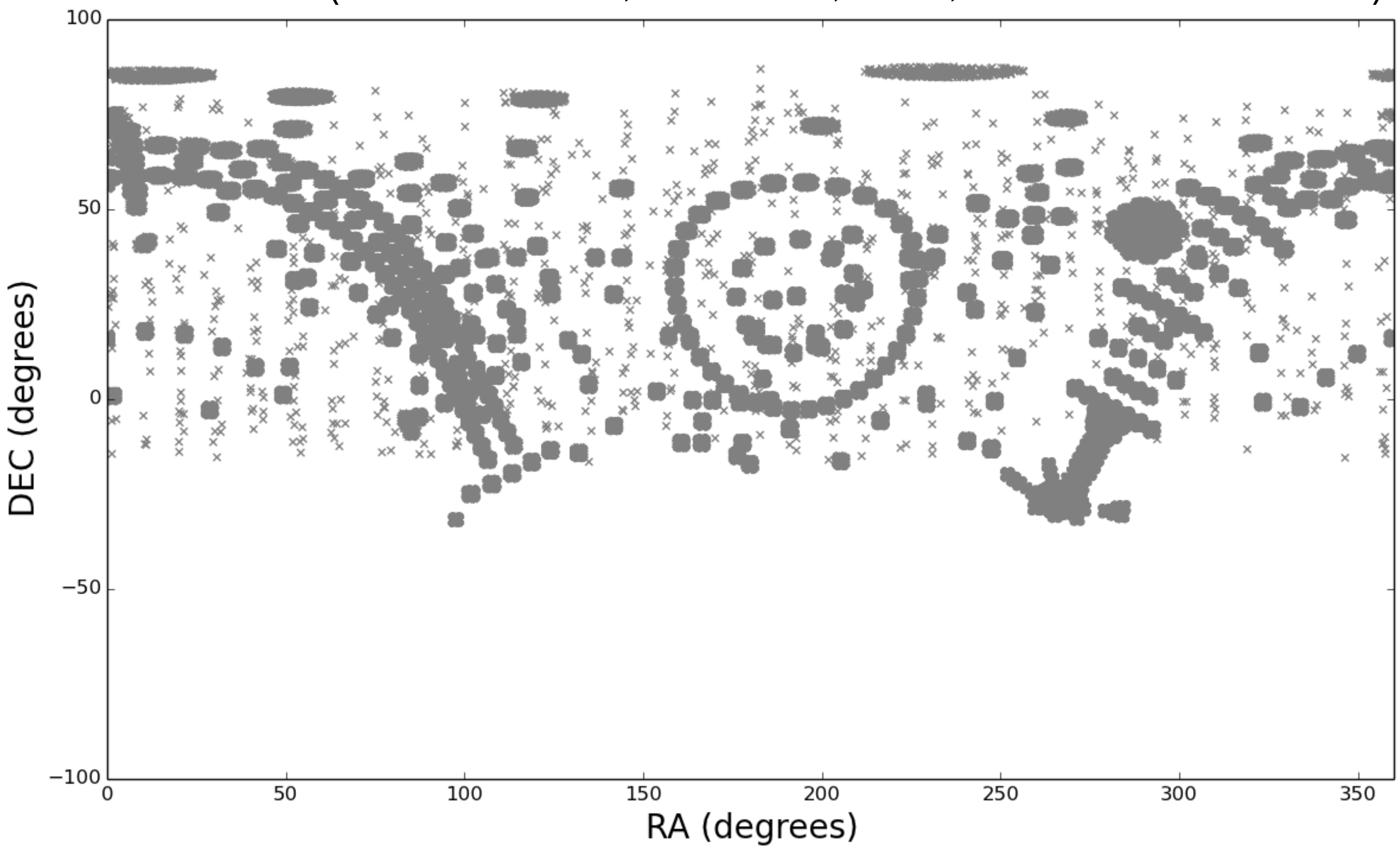
Bigger picture

- Multitude of surveys at different R, λ
- Independent pipelines for parameters

Bigger picture

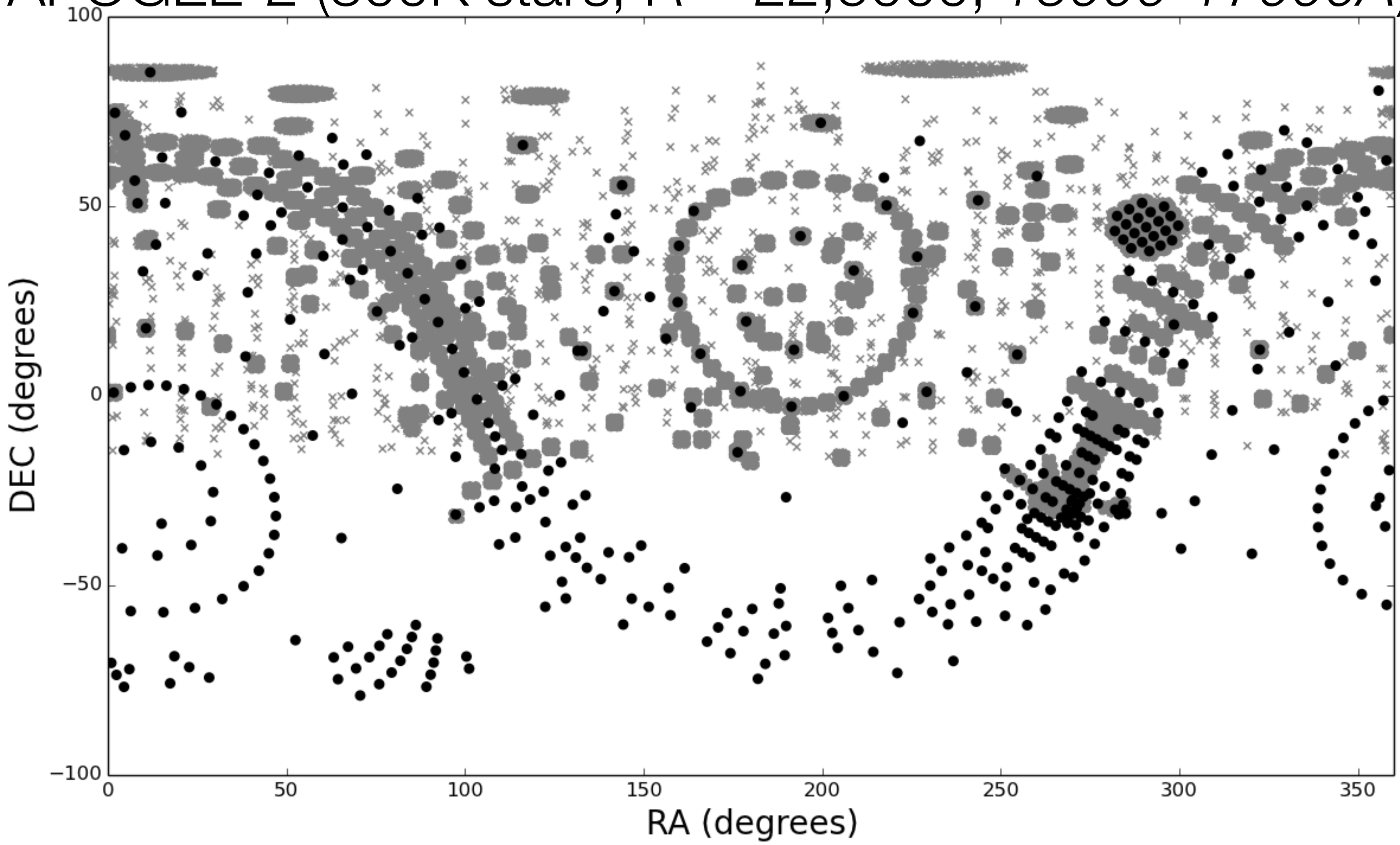
- Multitude of surveys at different R, λ
- Independent pipelines for parameters

APOGEE (150K stars, R = 22,500, *15000-17000A*)

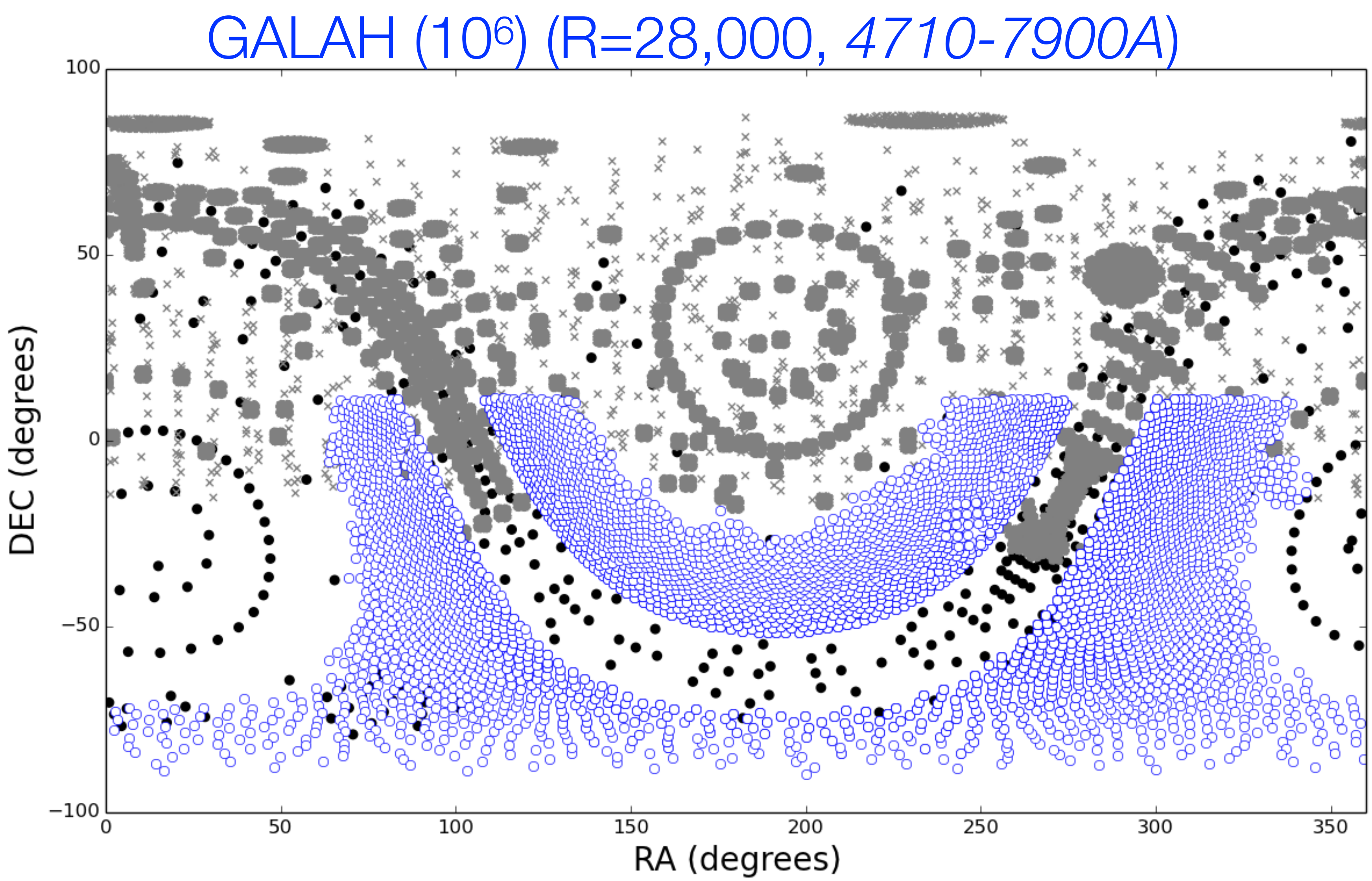


Bigger picture

APOGEE-2 (300K stars, R = 22,5000, *15000-17000A*)

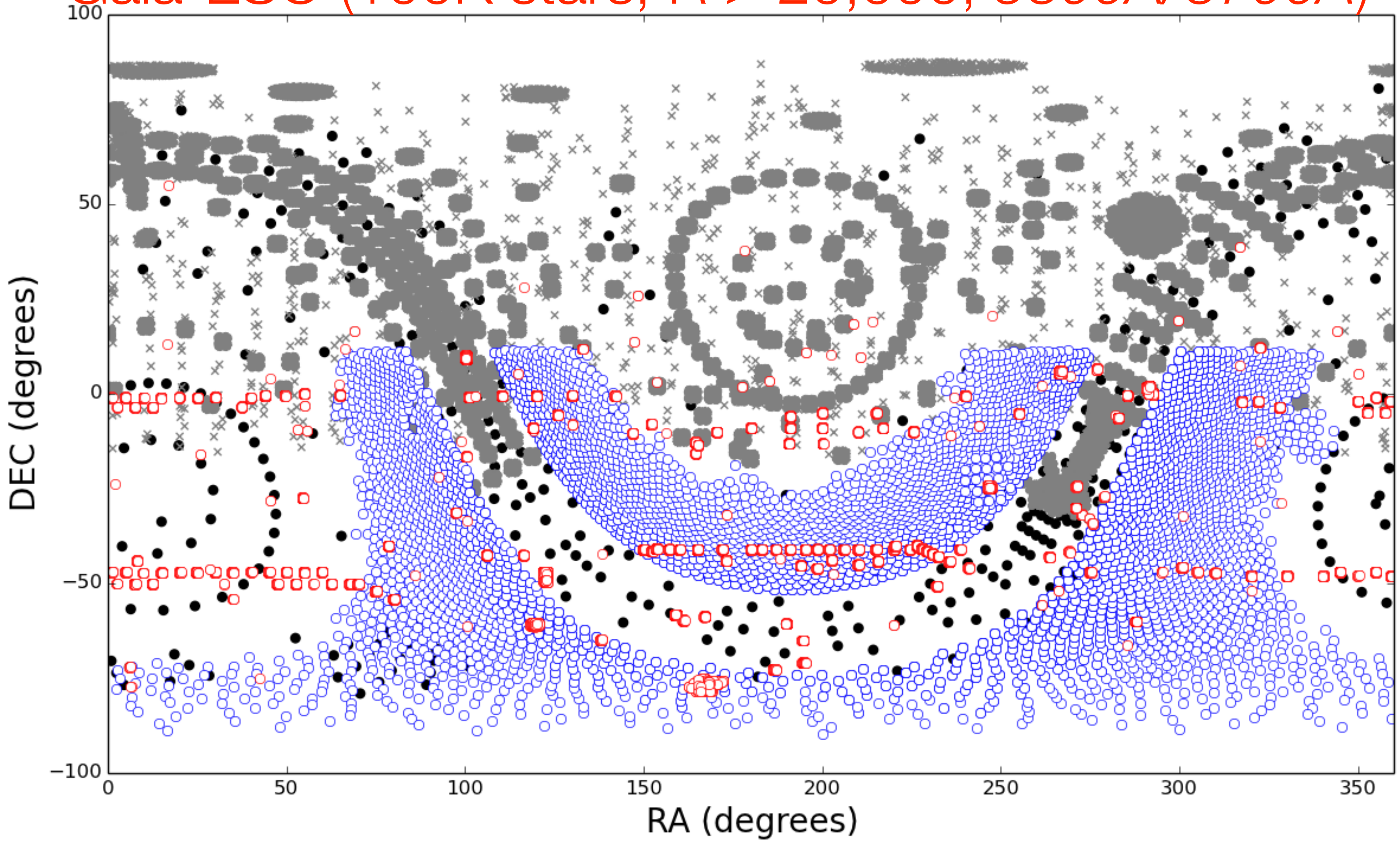


Bigger picture



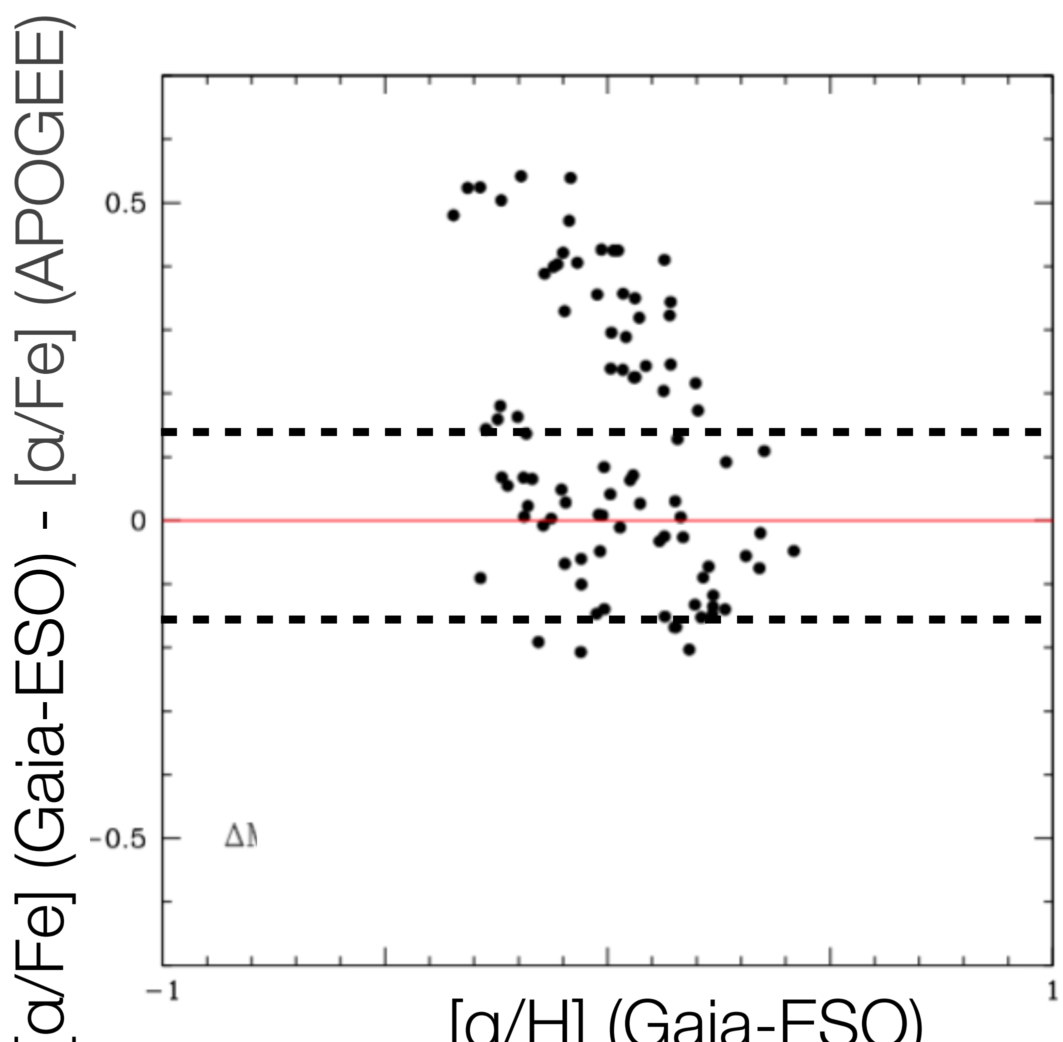
Bigger picture

Gaia-ESO (100K stars, R > 20,000, *5800A/8700A*)



The problem:

Different surveys have different scales for Teff, logg, [Fe/H], [X/Fe]



The solution

Sub-set of reference objects: stellar parameters, abundances (labels) known with high(er) fidelity.

- astroseismology
- clusters
- high SNR, well studied benchmarks, (e.g Jofre+ 2014)

The solution

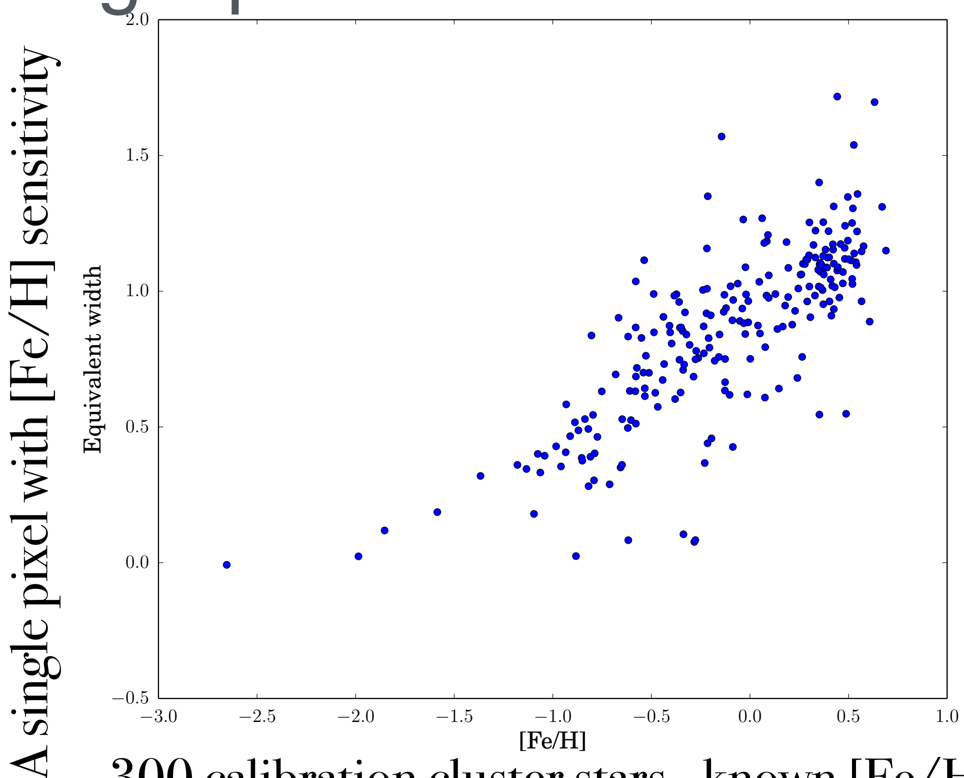
Sub-set of reference objects: stellar parameters, abundances (labels) known with high(er) fidelity.

- astroseismology
- clusters
- high SNR, well studied benchmarks, (e.g Jofre+ 2014)

use these labels *directly*: data-driven model

A very simple empirical model

A single pixel is information-rich

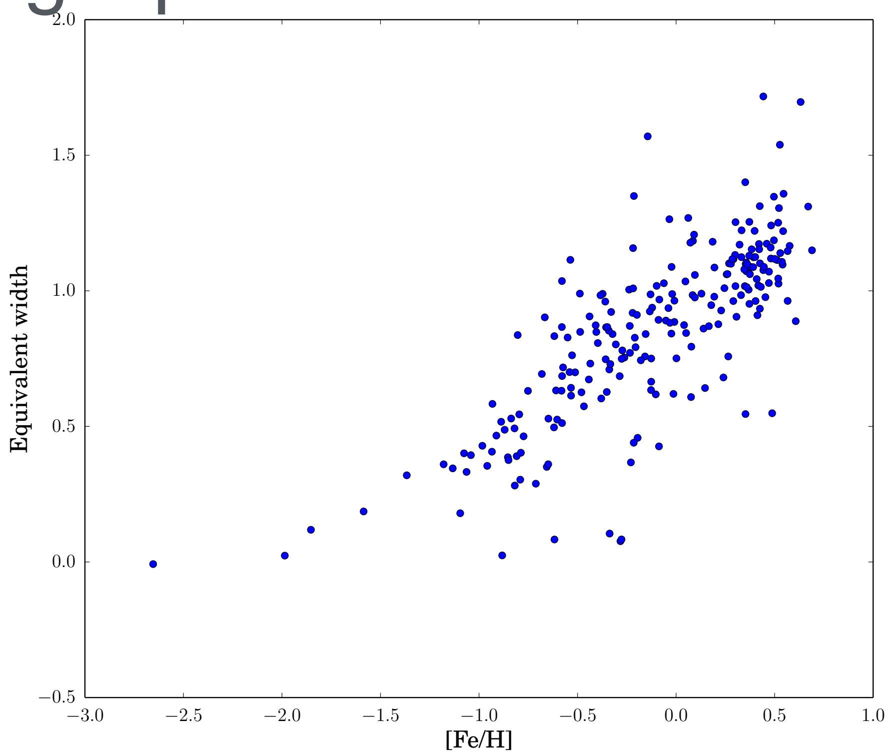


300 calibration cluster stars - known [Fe/H]

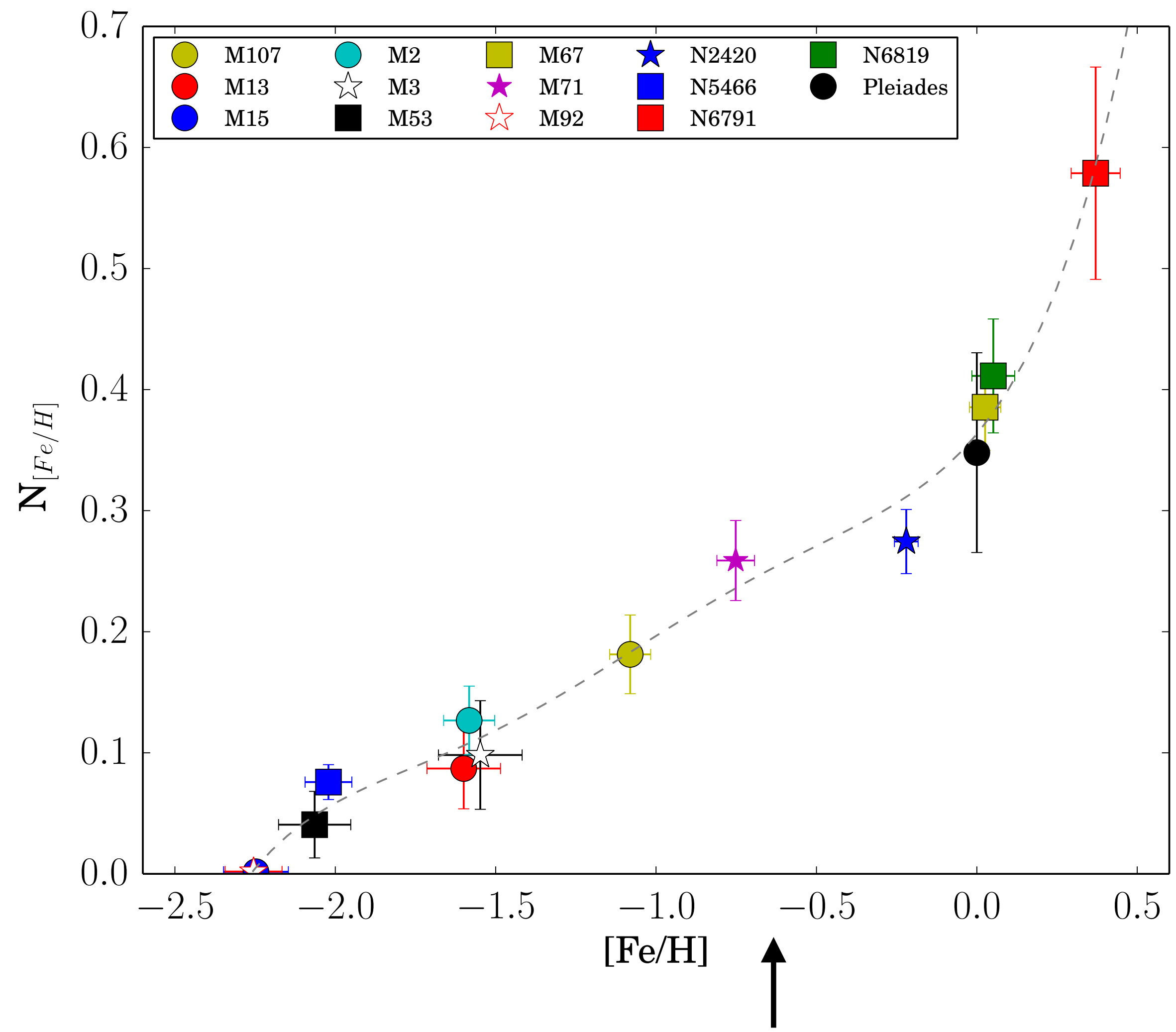
A very simple empirical model

A single pixel is information-rich

A single pixel with [Fe/H] sensitivity

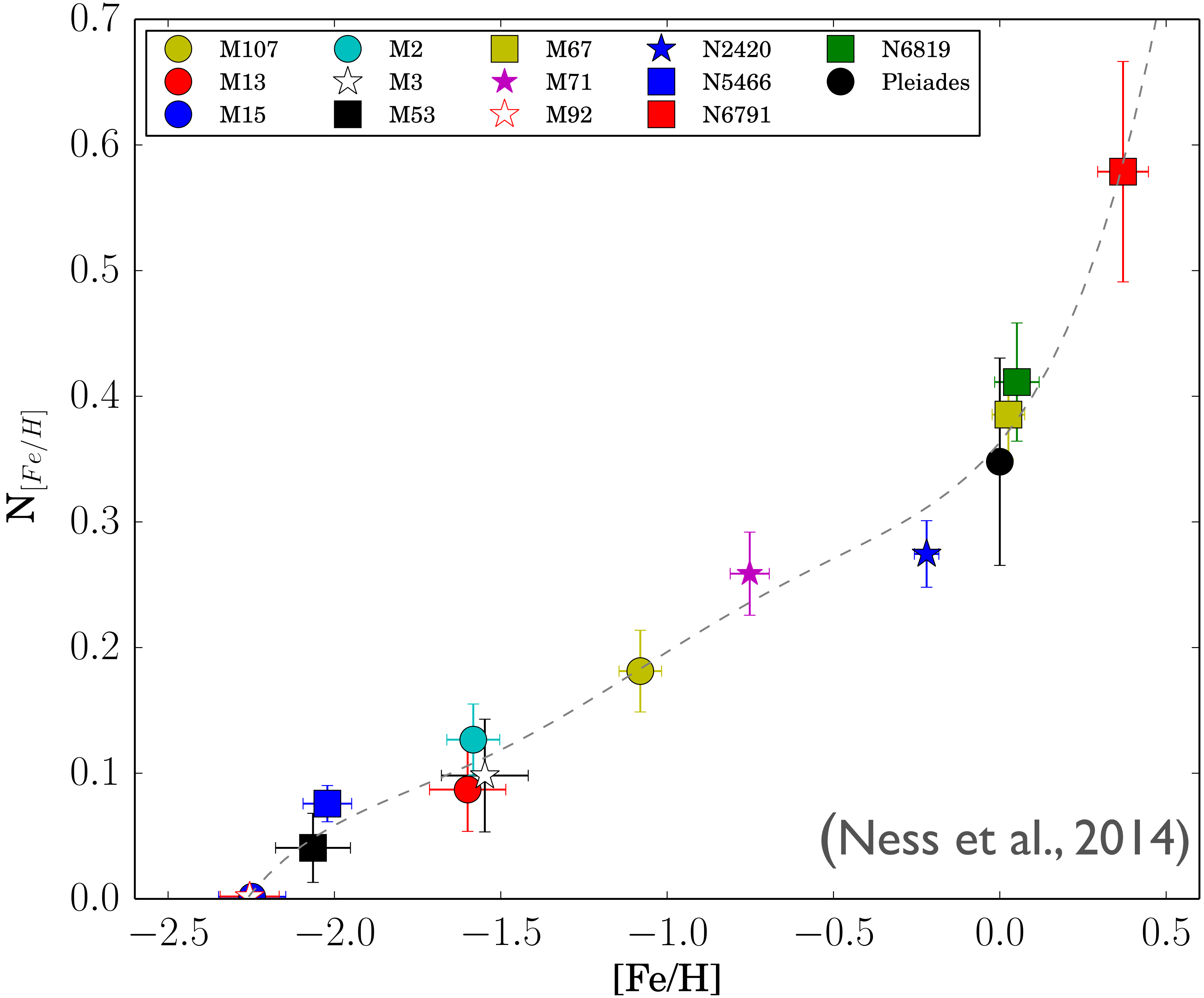


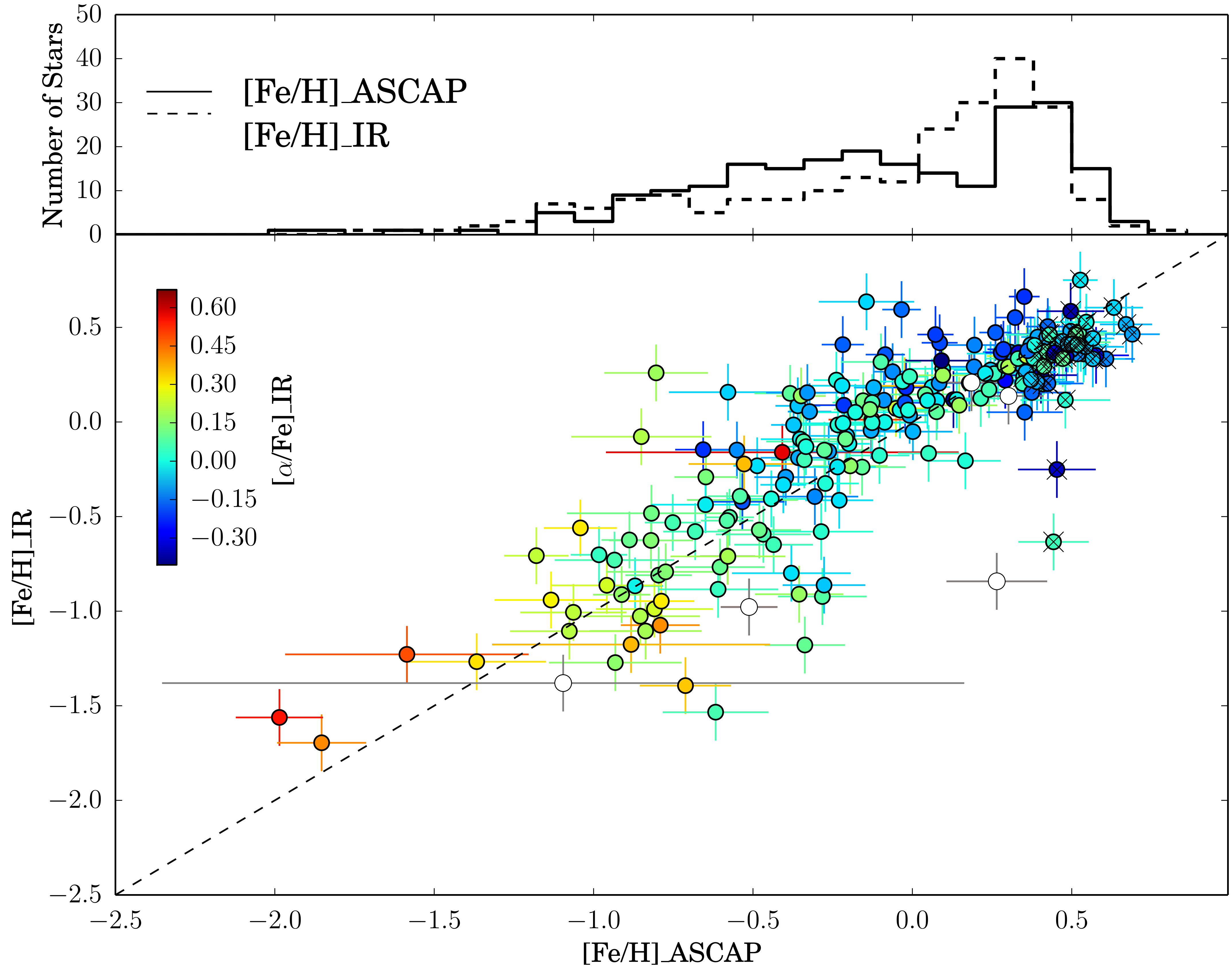
300 calibration cluster stars - known [Fe/H]



proof of concept: an index made (measuring EW of 5 features) using **1%** of the spectrum can use open and globular clusters to calibrate this index to be a precise [Fe/H] measure

5 x Features (30 Angstroms of spectra) metallicity sensitive features (log g, Teff insensitive)



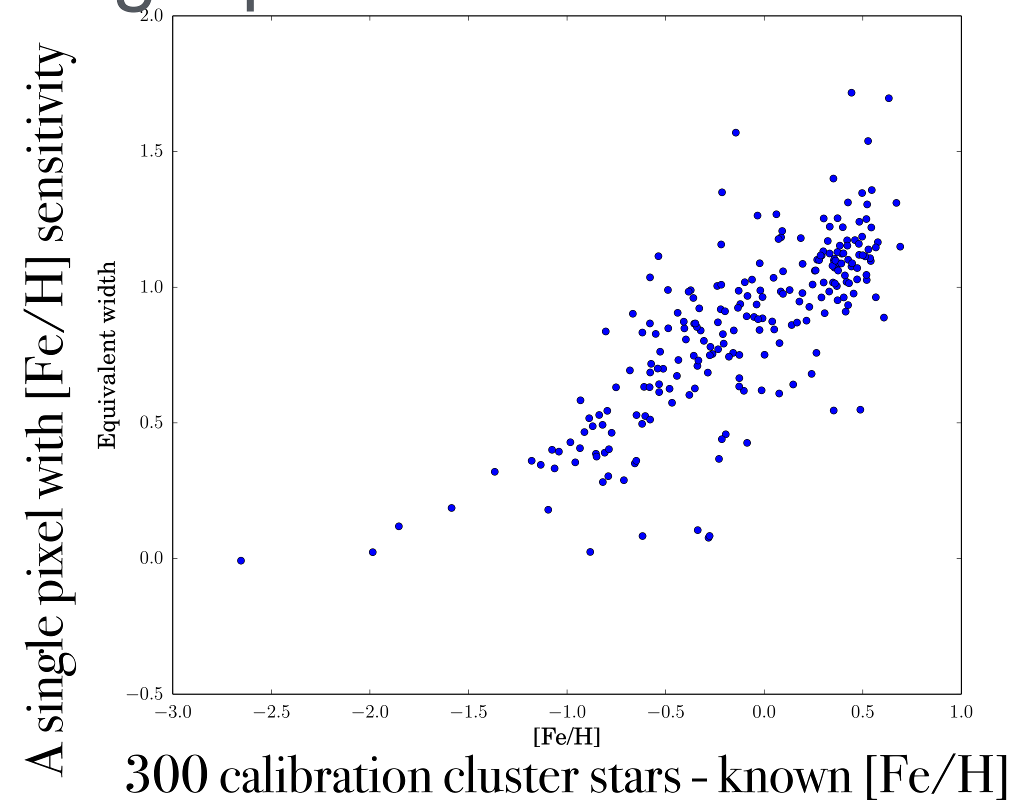


Number of Stars
[Fe/H]_ASCAP
[Fe/H]_IR
[α/Fe]_IR
0.60
0.45
0.30
0.15
0.00
−0.15
−0.30

The Cannon

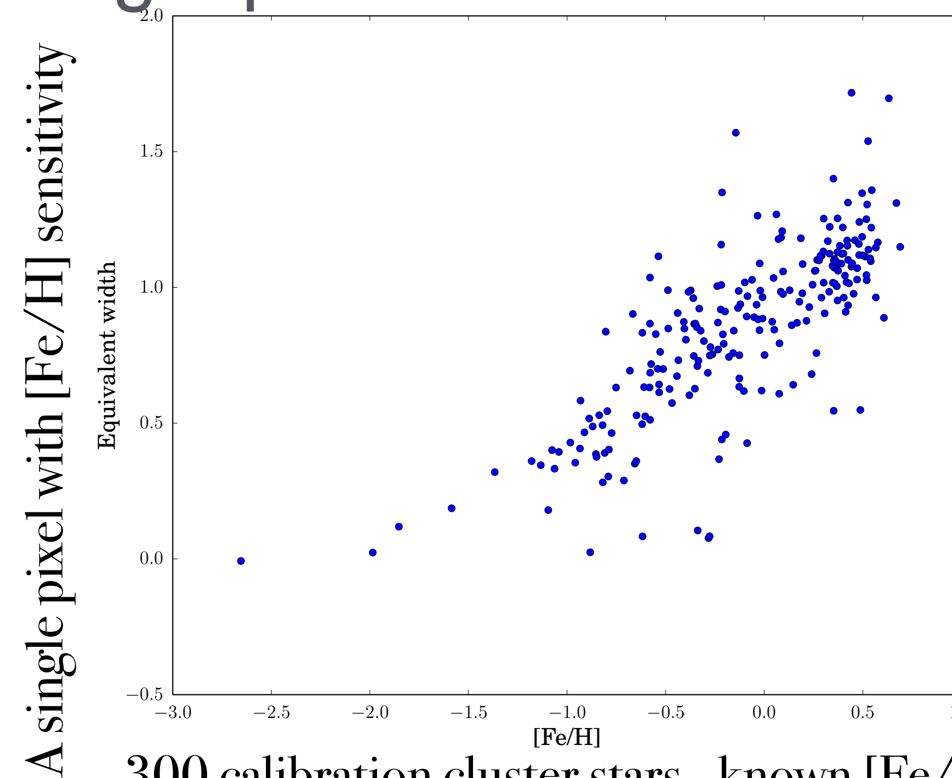
The Cannon

A single pixel is information-rich



The Cannon

A single pixel is information-rich

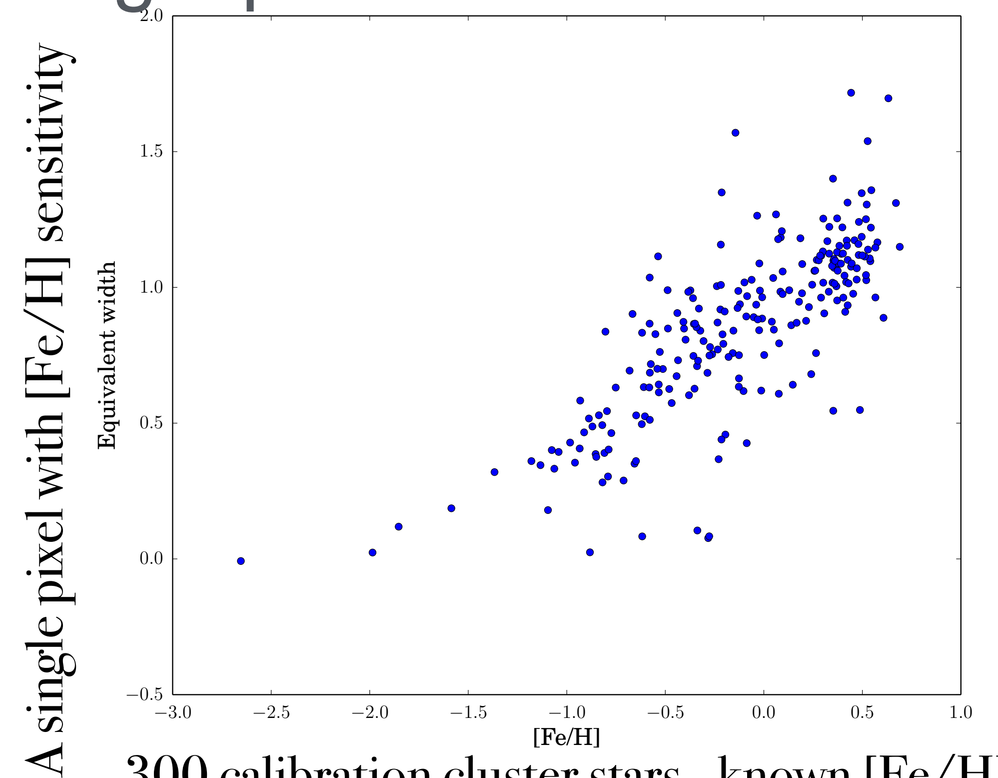


300 calibration cluster stars - known [Fe/H]

data-driven approach to measuring stellar parameters & abundances (labels) for stars in large surveys from the spectra directly

The Cannon

A single pixel is information-rich



300 calibration cluster stars - known [Fe/H]

data-driven approach to measuring stellar parameters & abundances (labels) for stars in large surveys from the spectra directly

http://arxiv.org/pdf/1501.07604v2.pdf

M. Ness, David W. Hogg, H.W. Rix, Anna Ho, & G. Zasowski

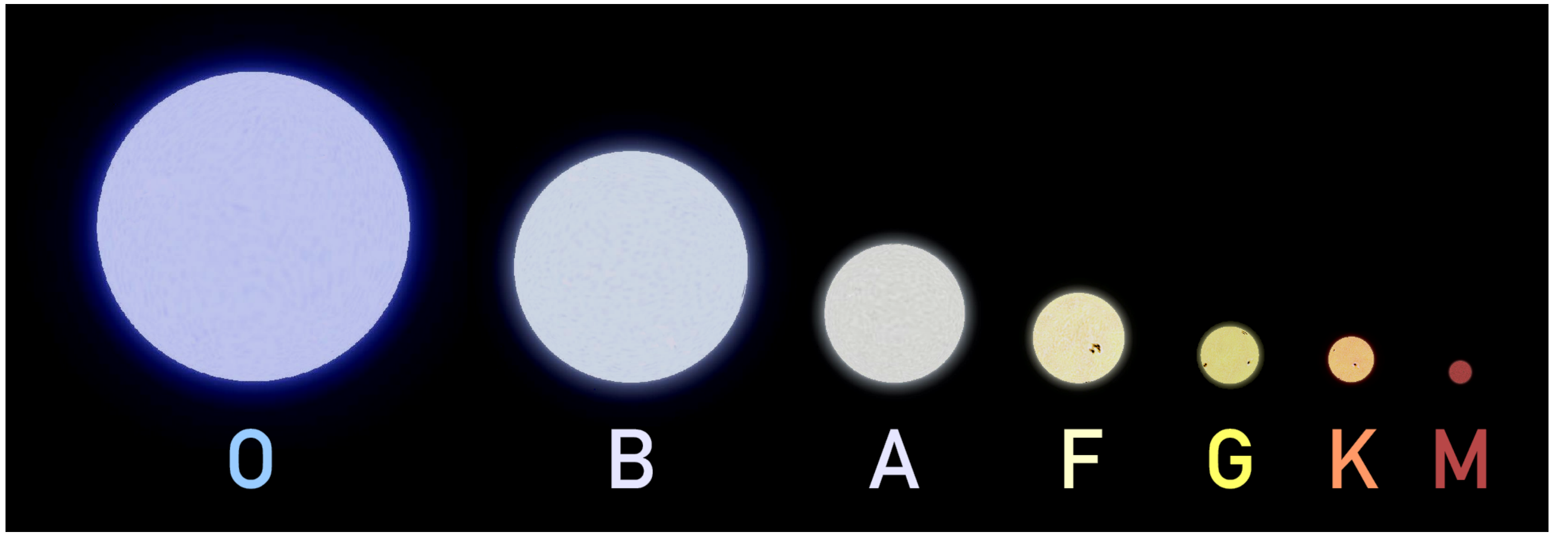
Annie Jump Cannon

Annie Jump Cannon

Characterized temperature sequence of stars
without stellar models

Annie Jump Cannon

Characterized temperature sequence of stars
without stellar models



The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

Uses reference objects (n) with known labels (ℓ_n) to build a model: *Training*

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

Uses reference objects (n) with known labels (ℓ_n) to build a model: *Training*

$$F_{n\lambda} = g(\ell_n; \theta_\lambda) + \text{error}$$

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

Uses reference objects (n) with known labels (ℓ_n) to build a model: *Training*

Teff, logg, [Fe/H]
[X/Fe]

$$F_{n\lambda} = g(\ell_n; \theta_\lambda) + \text{error}$$

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

Uses reference objects (n) with known labels (ℓ_n) to build a model: *Training*

Teff, logg, [Fe/H]
[X/Fe]

$$F_{n\lambda} = g(\ell_n; \theta_\lambda) + \text{error}$$

Spectral model params

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

Uses reference objects (n) with known labels (ℓ_n) to build a model: *Training*

Teff, logg, [Fe/H]
[X/Fe]

measurement error +
Inaccuracy of model

$$F_{n\lambda} = g(\ell_n; \theta_\lambda) + \text{error}$$

Spectral model params

The Basic Idea

Relies on a *subset* of reference stars in the survey, with known labels (Teff, logg, [Fe/H].)…

(i) stars with the same labels have the same spectra
(ii) flux varies smoothly with labels

Uses reference objects (n) with known labels (ℓ_n) to build a model: *Training*

Teff, logg, [Fe/H] [X/Fe]

measurement error + Inaccuracy of model

$$F_{n\lambda} = g(\ell_n; \theta_\lambda) + \text{error}$$

Spectral model params

That model is then used to infer the stellar labels for the remaining stars in the survey: *Test*

How about a linear model?

How about a linear model?

$$f_{n\lambda} = \quad a_\lambda + b_\lambda(\mathrm{Teff})_n + c_\lambda(\mathrm{logg})_n + d_\lambda([\mathrm{Fe/H}])_n + \quad \text{noise}$$

$$\theta_\lambda = (a_\lambda, b_\lambda, c_\lambda, d_\lambda)$$

How about a linear model?

(i) *Training step:* with the known labels: Teff, logg, [Fe/H]

$$f_{n\lambda} = \quad a_\lambda + b_\lambda(\mathrm{Teff})_n + c_\lambda(\mathrm{logg})_n + d_\lambda([\mathrm{Fe/H}])_n + \quad \text{noise}$$

$$\theta_\lambda = (a_\lambda, b_\lambda, c_\lambda, d_\lambda)$$

How about a linear model?

(i) *Training step:* with the known labels: Teff, logg, [Fe/H]

$$f_{n\lambda} = \quad a_\lambda + b_\lambda(\mathrm{Teff})_n + c_\lambda(\mathrm{logg})_n + d_\lambda([\mathrm{Fe/H}])_n + \quad \mathrm{noise}$$

$$\theta_\lambda = (a_\lambda, b_\lambda, c_\lambda, d_\lambda)$$

(i) *Test step:* with the survey stars

How about a linear model?

(i) *Training step:* with the known labels: Teff, logg, [Fe/H]

$$f_{n\lambda} = a_\lambda + b_\lambda(\mathrm{Teff})_n + c_\lambda(\mathrm{logg})_n + d_\lambda([\mathrm{Fe/H}])_n + \text{noise}$$

$\theta_\lambda = (a_\lambda, b_\lambda, c_\lambda, d_\lambda)$

(i) *Test step:* with the survey stars

$$f_{m\lambda} = \mathrm{a}_\lambda + \mathrm{b}_\lambda(\mathrm{Teff})_m + \mathrm{c}_\lambda(\mathrm{logg})_m + \mathrm{d}_\lambda([\mathrm{Fe/H}])_m + \text{noise}_\lambda$$

How about a linear model?

(i) *Training step:* with the known labels: Teff, logg, [Fe/H]

$$f_{n\lambda} = a_\lambda + b_\lambda(\mathrm{Teff})_n + c_\lambda(\mathrm{logg})_n + d_\lambda([\mathrm{Fe/H}])_n + \text{noise}$$

$\theta_\lambda = (a_\lambda, b_\lambda, c_\lambda, d_\lambda)$

(i) *Test step:* with the survey stars

$$f_{m\lambda} = \mathrm{a}_\lambda + \mathrm{b}_\lambda(\mathrm{Teff})_m + \mathrm{c}_\lambda(\mathrm{logg})_m + \mathrm{d}_\lambda([\mathrm{Fe/H}])_m + \text{noise}_\lambda$$

Unfortunately, this is too simple

We also need products of the labels

We also need products of the labels

$$\eta\left(\begin{bmatrix} T_{\mathrm{eff}} \\ \log g \\ [X/\mathrm{Fe}] \end{bmatrix}\right) = \begin{bmatrix} 1 \\ T_{\mathrm{eff}} \\ \log g \\ [X/\mathrm{Fe}] \\ (T_{\mathrm{eff}})^2 \\ (\log g)^2 \\ ([X/\mathrm{Fe}])^2 \\ (T_{\mathrm{eff}})(\log g) \\ (T_{\mathrm{eff}})([X/\mathrm{Fe}]) \\ (\log g)([X/\mathrm{Fe}]) \end{bmatrix}$$

We also need products of the labels

$$\eta\left(\begin{bmatrix} T_{\text{eff}} \\ \log g \\ [X/\text{Fe}] \end{bmatrix}\right) = \begin{bmatrix} 1 \\ T_{\text{eff}} \\ \log g \\ [X/\text{Fe}] \\ (T_{\text{eff}})^2 \\ (\log g)^2 \\ ([X/\text{Fe}])^2 \\ (T_{\text{eff}})(\log g) \\ (T_{\text{eff}})([X/\text{Fe}]) \\ (\log g)([X/\text{Fe}]) \end{bmatrix}$$

Pivot the labels:

$$T'_{\text{eff}} = \frac{T_{\text{eff}} - \overline{T_{\text{eff}}}}{\text{std}(T_{\text{eff}})}$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix}$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\mathrm{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\mathrm{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix}$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad F_{n\lambda} = g(\ell_n; \theta_\lambda) + \text{error}$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

$$\begin{aligned}
F_{n\lambda} = \theta_0 & \qquad \text{(constant term)} \\
+ \theta_{T_{\text{eff}}} T_{\text{eff}} + \cdots + \theta_{X_N} [X_N/\text{Fe}] & \qquad \text{(linear terms)} \\
+ \theta_{T_{\text{eff}}^2} T_{\text{eff}}^2 + \cdots + \theta_{X_N^2} ([X_N/\text{Fe}])^2 & \qquad \text{(squared terms)} \\
+ \theta_{T_{\text{eff}} \log(g)} T_{\text{eff}} \log(g) + \ldots & \\
+ \theta_{X_N X_{N-1}} [X_N/\text{Fe}][X_{N-1}/\text{Fe}] & \qquad \text{(cross-terms)} \\
+ \text{error} &
\end{aligned}$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix}$$

$$\begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

$$\begin{aligned} F_{n\lambda} = \theta_0 & & \text{(constant term)} \\ + \theta_{T_{\text{eff}}} T_{\text{eff}} + \cdots + \theta_{X_N} [X_N/\text{Fe}] & & \text{(linear terms)} \\ + \theta_{T^2_{\text{eff}}} T^2_{\text{eff}} + \cdots + \theta_{X^2_N} ([X_N/\text{Fe}])^2 & & \text{(squared terms)} \\ + \theta_{T_{\text{eff}} \log(g)} T_{\text{eff}} \log(g) + \ldots & & \\ + \theta_{X_N X_{N-1}} [X_N/\text{Fe}][X_{N-1}/\text{Fe}] & & \text{(cross-terms)} \\ + \text{error} & & \end{aligned}$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\rm eff})_n \\ (\log g)_n \\ \vdots \\ ([X/{\rm Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

Get specific about "error"

$$(\text{error})^2 = (\text{measurement error})^2 + (\text{unmodelled effects})^2$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

Get specific about "error"

$$(\text{error})^2 = (\text{measurement error})^2 + (\text{unmodelled effects})^2$$

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

Get specific about "error"

$$(\text{error})^2 = (\text{measurement error})^2 + (\text{unmodelled effects})^2$$

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

Get specific about "error"

$$(\text{error})^2 = (\text{measurement error})^2 + (\text{unmodelled effects})^2$$

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma^2_{n\lambda} + s^2_\lambda)$$

Write down the model…

$$\ell_n = \begin{bmatrix} (T_{\text{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\text{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \text{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \text{error} \end{aligned}$$

Get specific about "error"

$$(\text{error})^2 = (\text{measurement error})^2 + (\text{unmodelled effects})^2$$

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma^2_{n\lambda} + s^2_\lambda)$$

Write down the model…

$$\boldsymbol{\ell}_n = \begin{bmatrix} (T_{\mathrm{eff}})_n \\ (\log g)_n \\ \vdots \\ ([X/\mathrm{Fe}])_n \end{bmatrix} \quad \theta_\lambda = \begin{bmatrix} a_\lambda \\ b_\lambda \\ c_\lambda \\ \vdots \end{bmatrix} \quad \begin{aligned} F_{n\lambda} &= g(\ell_n; \theta_\lambda) + \mathrm{error} \\ &= \theta_\lambda \cdot \eta(\ell_n) + \mathrm{error} \end{aligned}$$

Get specific about "error"

$$(\mathrm{error})^2 = (\mathrm{measurement\ error})^2 + (\mathrm{unmodelled\ effects})^2$$

$$\boxed{F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)}$$

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)$$

$$F_{n\lambda}|\boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)$$

Now training (test) is merely a matter of optimizing the sum of log likelihoods for each pixel (star)

$$F_{n\lambda}|\boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)$$

Now training (test) is merely a matter of optimizing the sum of log likelihoods for each pixel (star)

The model is *linear with gaussian error* in theta!
(for fixed s, ell)

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma^2_{n\lambda} + s^2_\lambda)$$

Now training (test) is merely a matter of optimizing the sum of log likelihoods for each pixel (star)

The model is *linear with gaussian error* in theta!
(for fixed s, ell)

$$F_{n\lambda}|\boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n)\cdot\boldsymbol{\theta}_\lambda, \sigma^2_{n\lambda} + s^2_\lambda)$$

If you want, apply priors (at either step), or regularize

Now training (test) is merely a matter of optimizing the sum of log likelihoods for each pixel (star)

The model is *linear with gaussian error* in theta!
(for fixed s, ell)

$$F_{n\lambda} | \boldsymbol{\ell}_n, \boldsymbol{\theta}_\lambda, s_\lambda \sim \mathcal{N}(\boldsymbol{\eta}(\boldsymbol{\ell}_n) \cdot \boldsymbol{\theta}_\lambda, \sigma_{n\lambda}^2 + s_\lambda^2)$$

If you want, apply priors (at either step), or regularize

The Cannon 2: A data-driven model of stellar spectra for detailed chemical abundance analyses

Andrew R. Casey[1], David W. Hogg[2,3,4,5], Melissa Ness[5], Hans-Walter Rix[5], Anna Y. Q. Ho[6], and Gerry Gilmore[1]

The Cannon in Action

Training set: 540 open and globular cluster stars, labels from ASPCAP (Teff, log g, [Fe/H])

Test set:

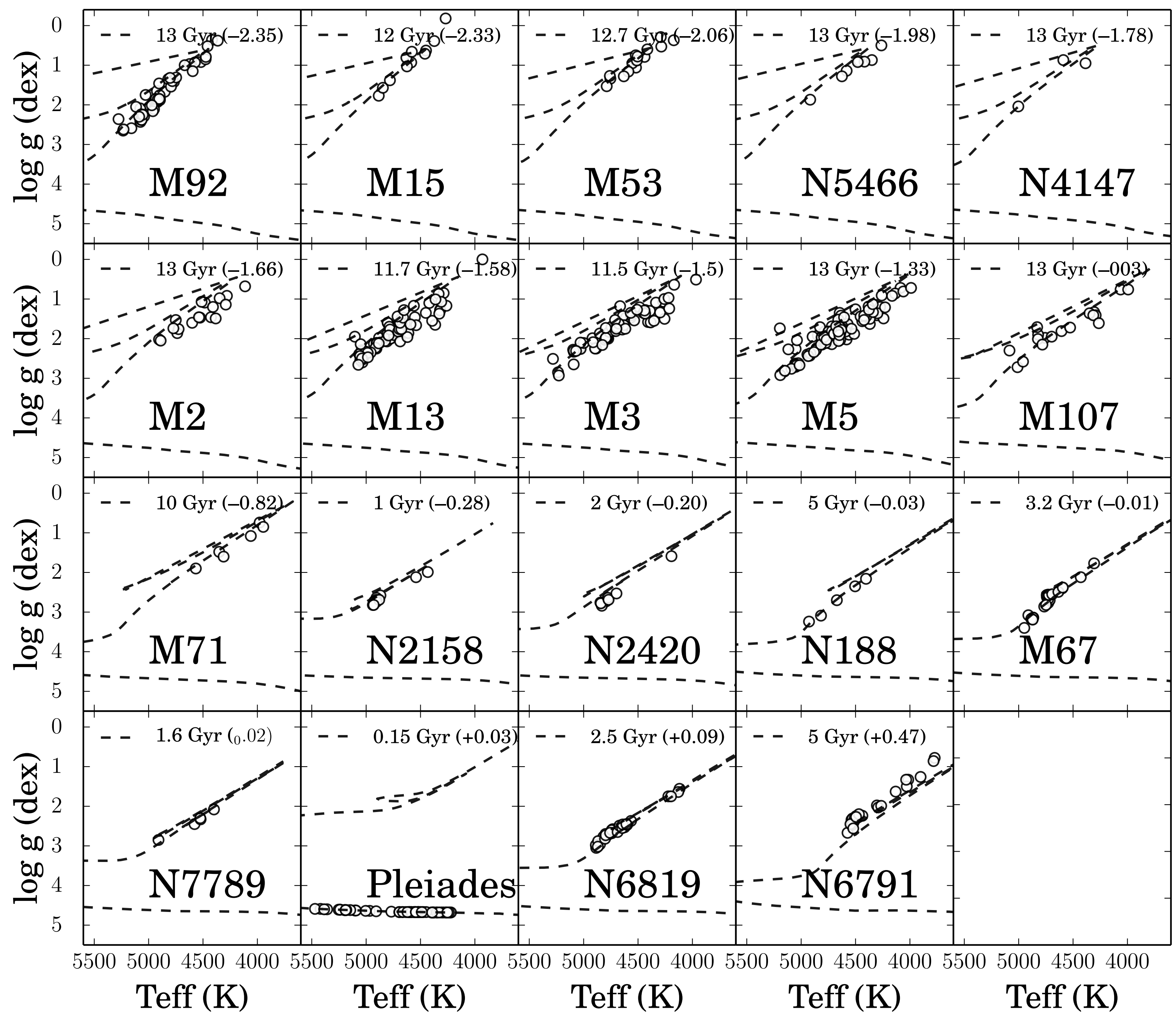
120,000 stars from APOGEE

The Cannon in Action

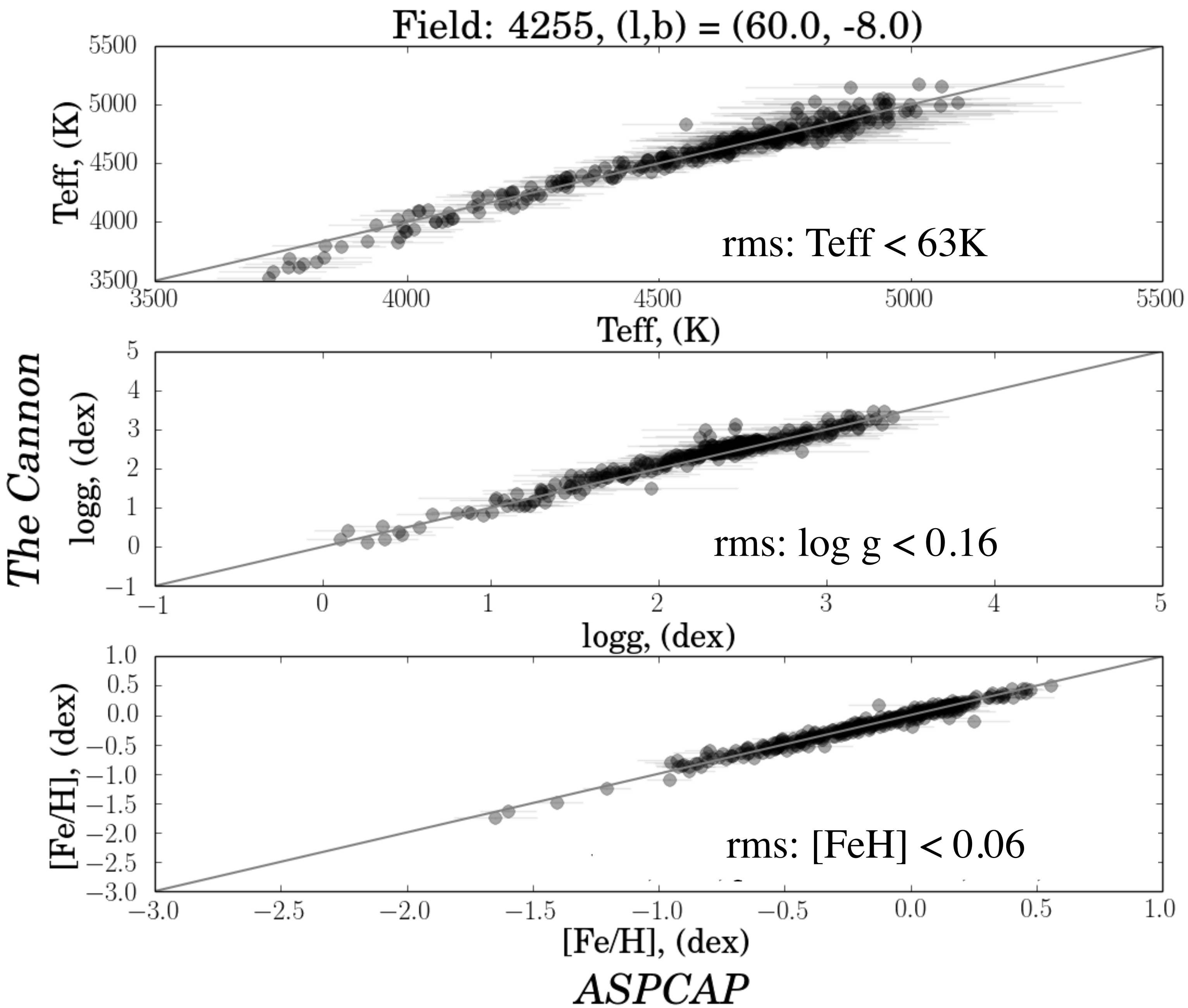
Training set: 540 open and globular cluster stars, labels from ASPCAP (Teff, log g, [Fe/H])

Test set:

120,000 stars from APOGEE



Cross-Validation

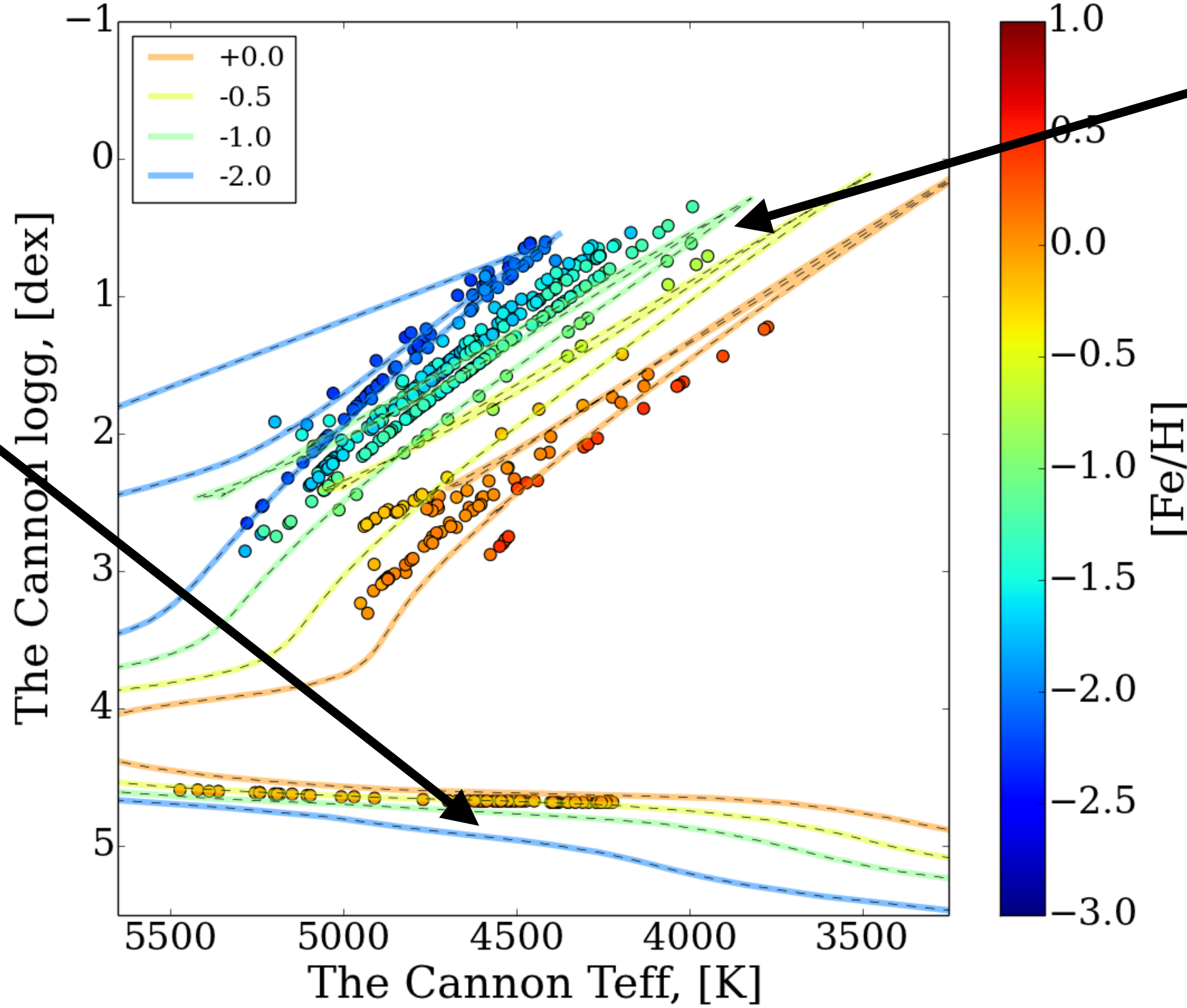


rms is the order of the ASPCAP errors — first indicator of high precision using this approach

APOGEE Results

Labels for 120,000 stars - 540 reference stars with high fidelity labels

only 1 cluster with main sequence stars amongst the reference objects



no isochrone priors

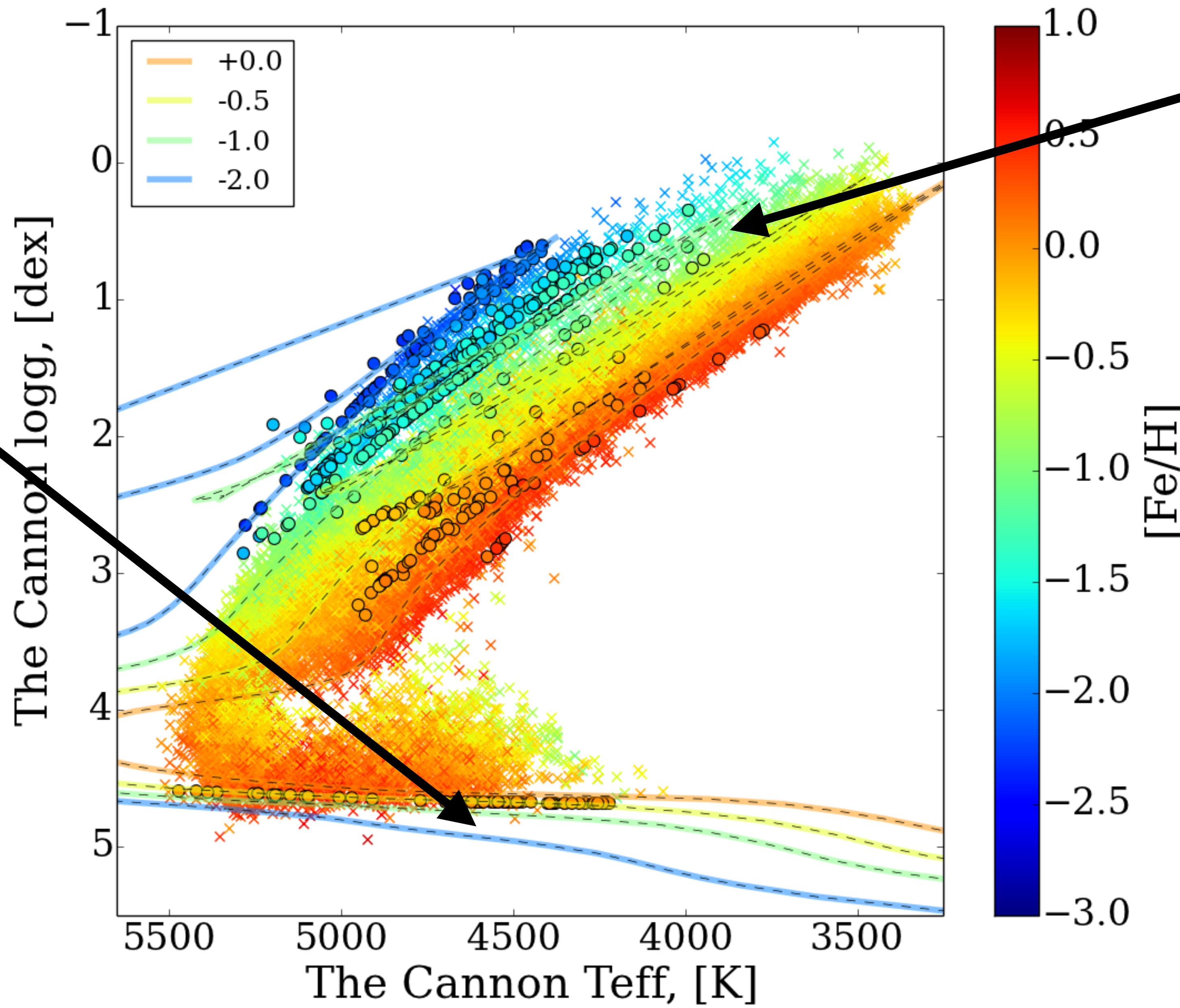
very fast!

APOGEE Results

Labels for 120,000 stars - 540 reference stars with high fidelity labels

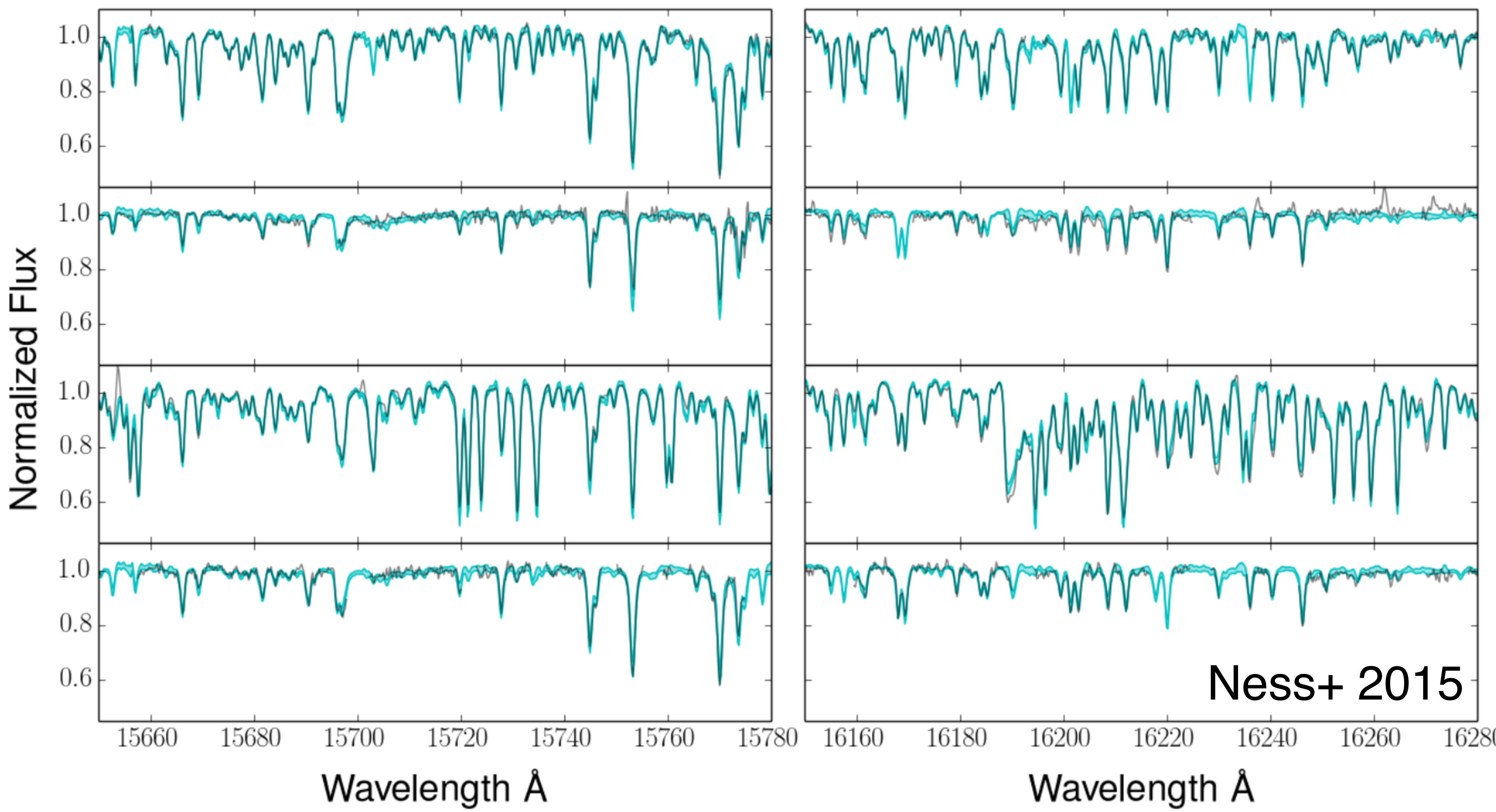
only 1 cluster with main sequence stars amongst the reference objects

no isochrone priors



very fast!

The model matches the data (3 labels!)



uses full spectrum & error-weighted information in each pixel

match becomes even better with abundance labels (see next example)

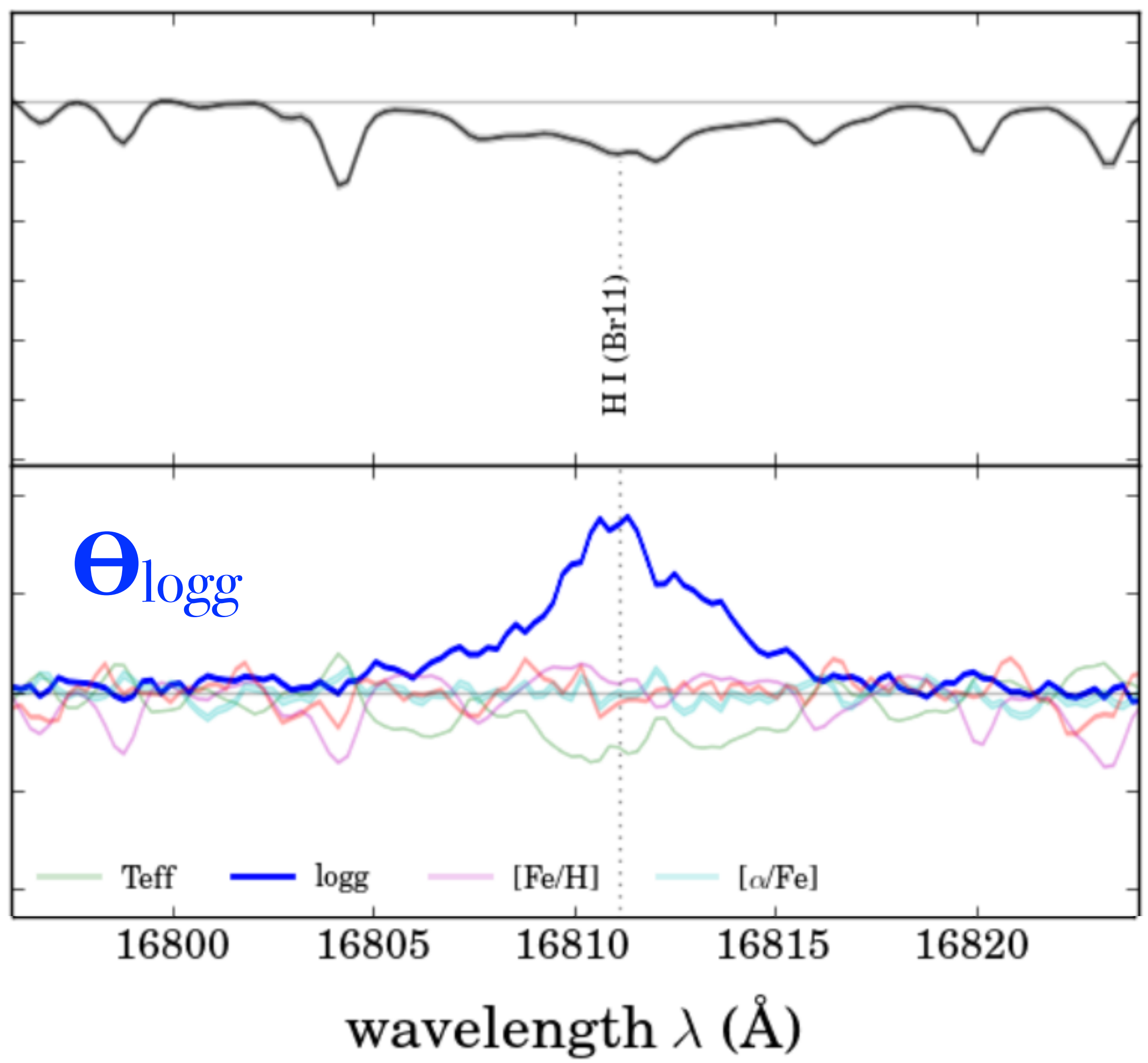
Where the information resides

$$F_{n\lambda} = \theta_\lambda \cdot \eta(\ell_n) + \text{error}$$

this line is where spectroscopists expect logg sensitivity from stellar physics - The Cannon learns this from the data ensemble (data-driven)

zeroth coefficient →

label coefficients →

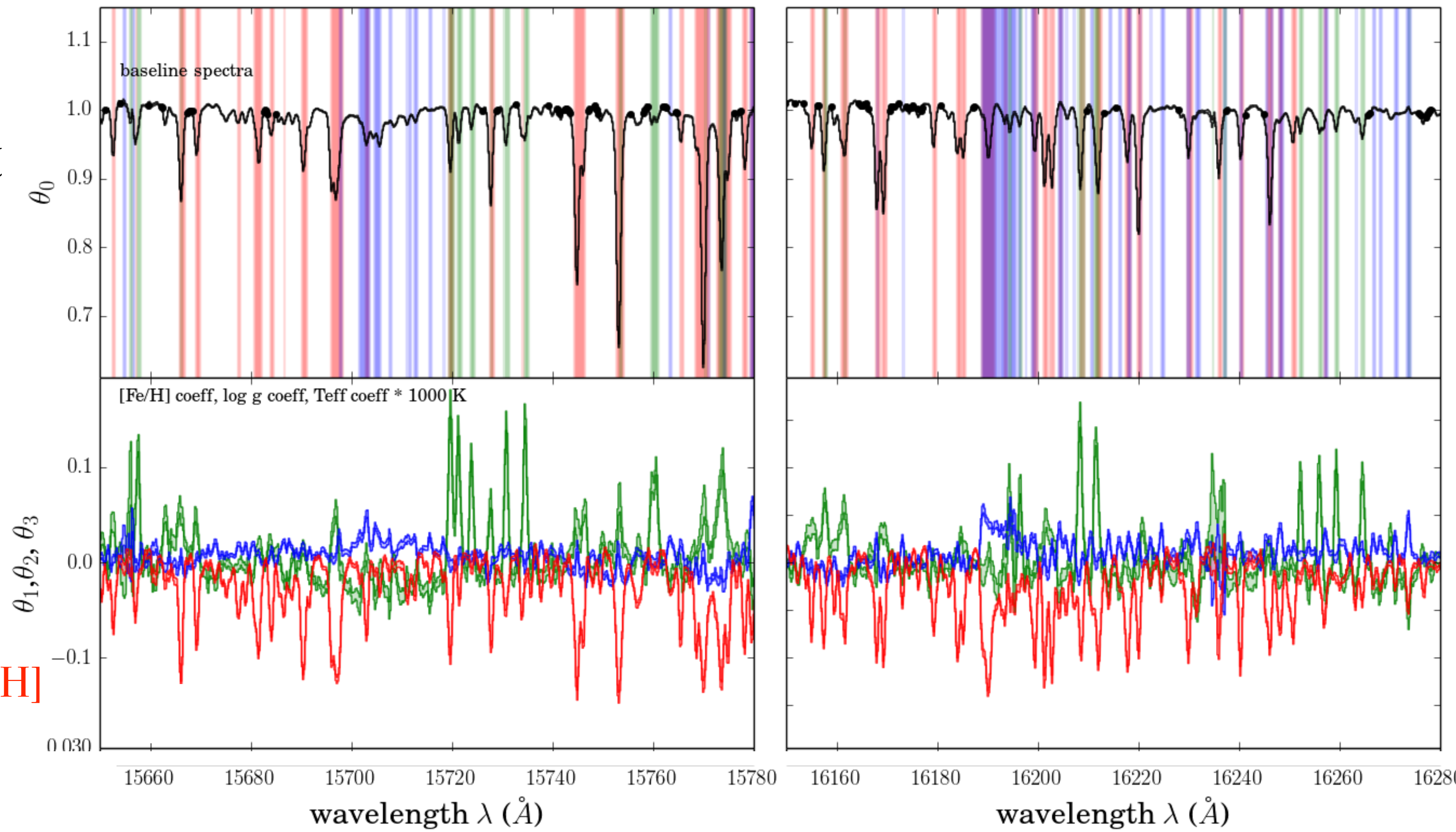


Where the information resides

zeroth coefficient →

label coefficients →

θ_{Teff}, θ_{logg}, $\theta_{[Fe/H]}$



Implementations

github.com/ajwheeler/TheCannon.jl (minimalmist)

github.com/andycasey/AnniesLasso (maximalist)

Applications…

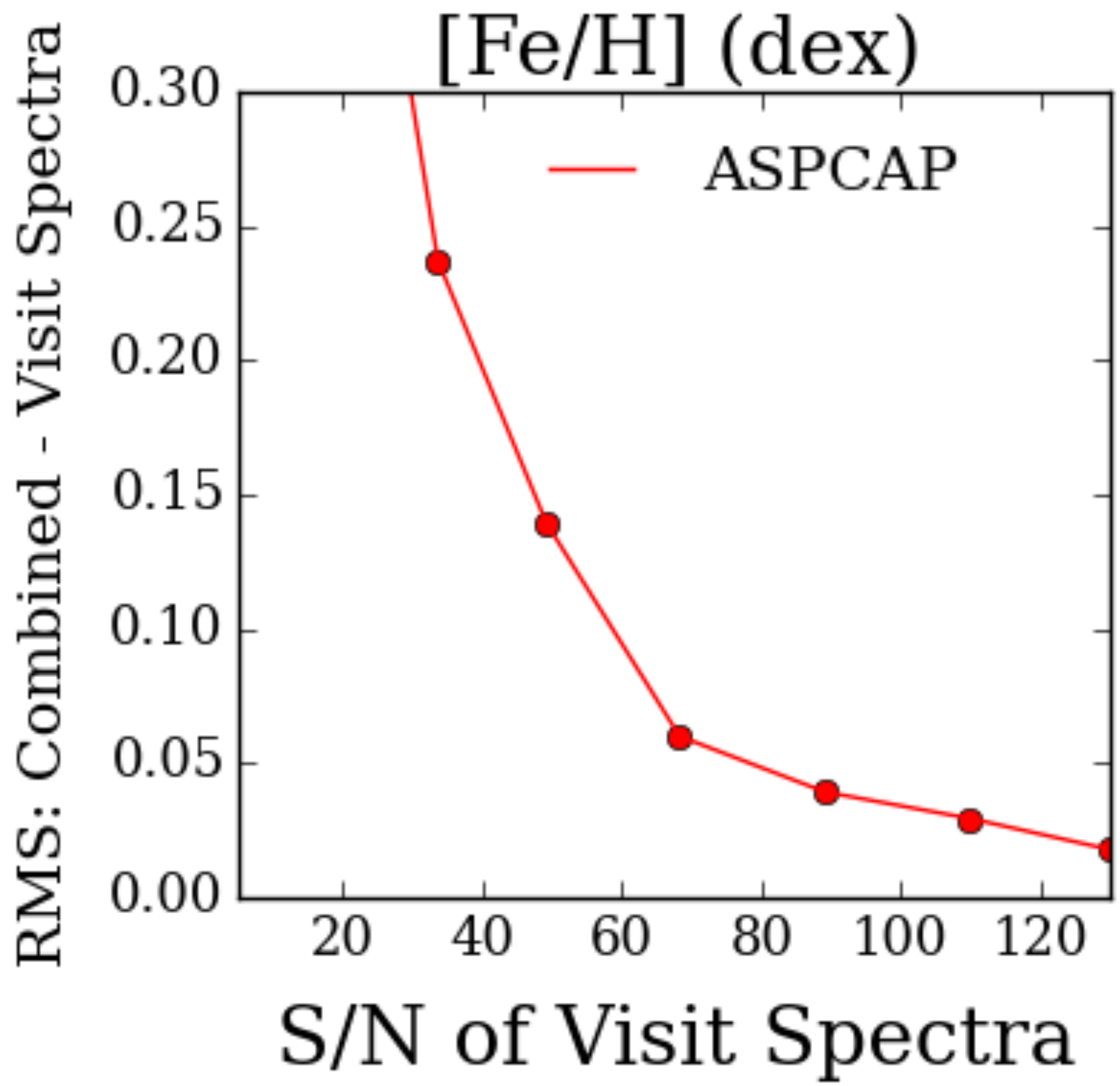
Precision abundances

Applications…

Precision abundances

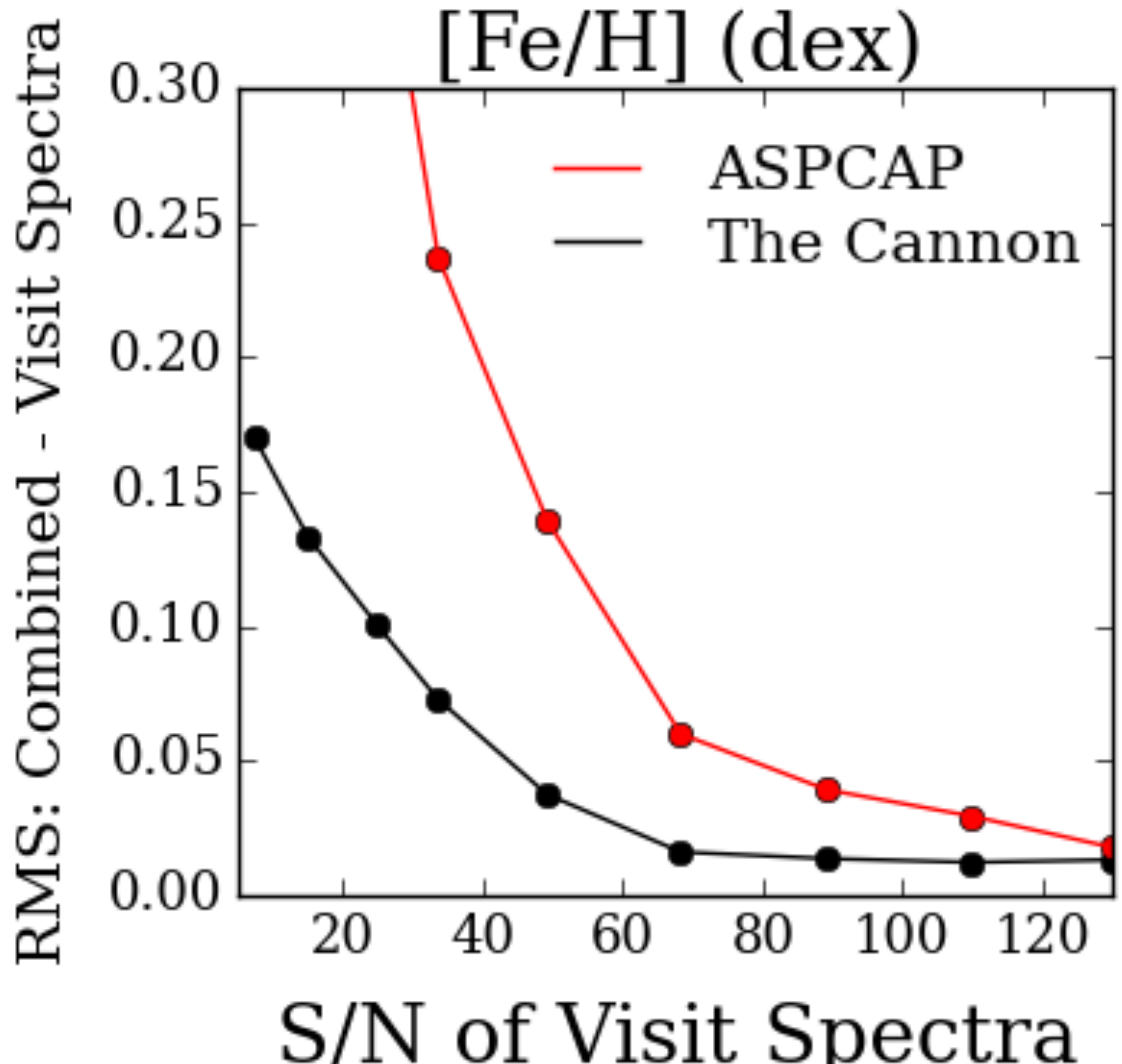
Advantages: The Cannon delivers 2-3 times smaller

The Cannon delivers 2-3 times smaller errors



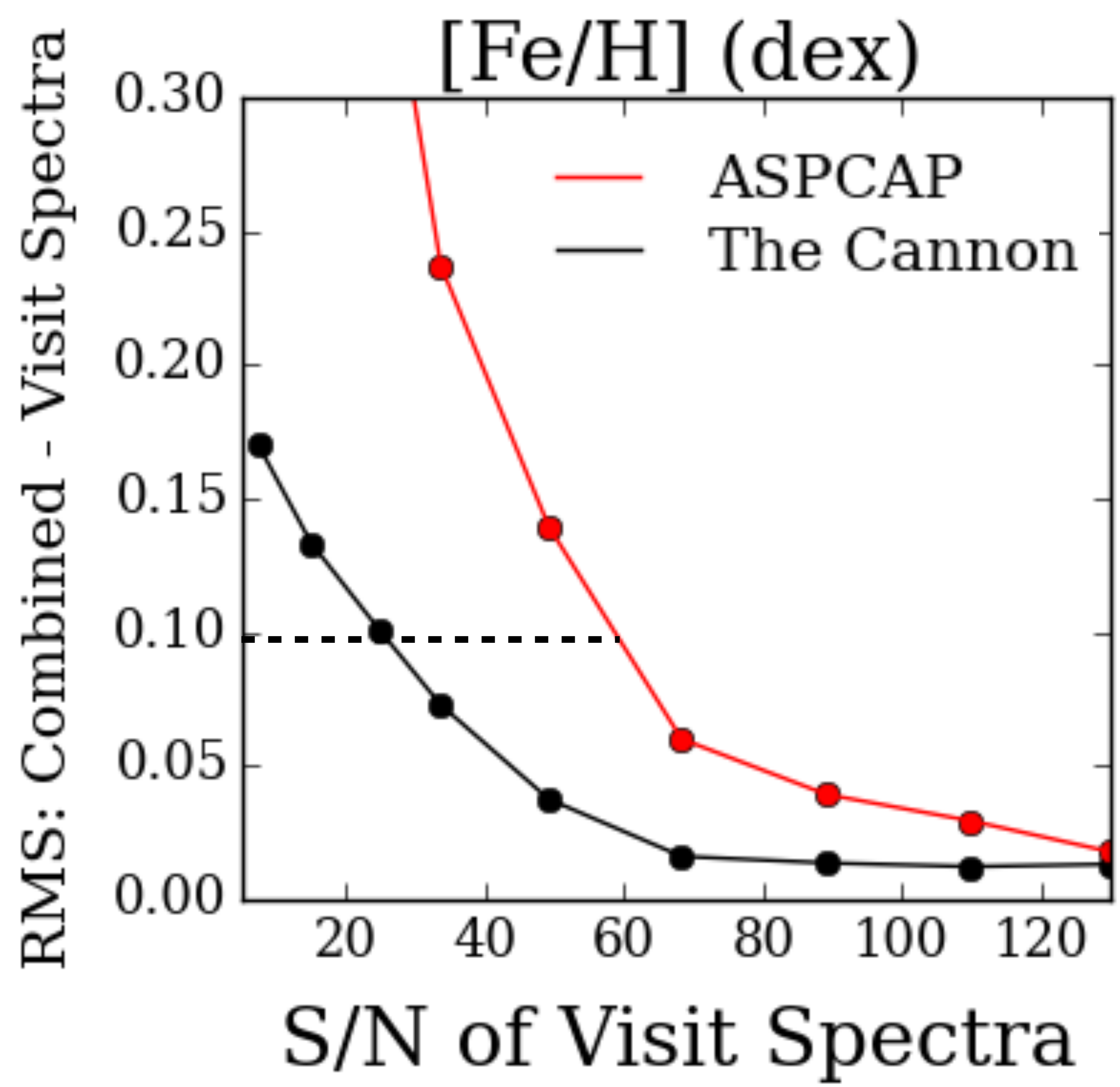
using multiple observations to assess precision

The Cannon delivers 2-3 times smaller errors



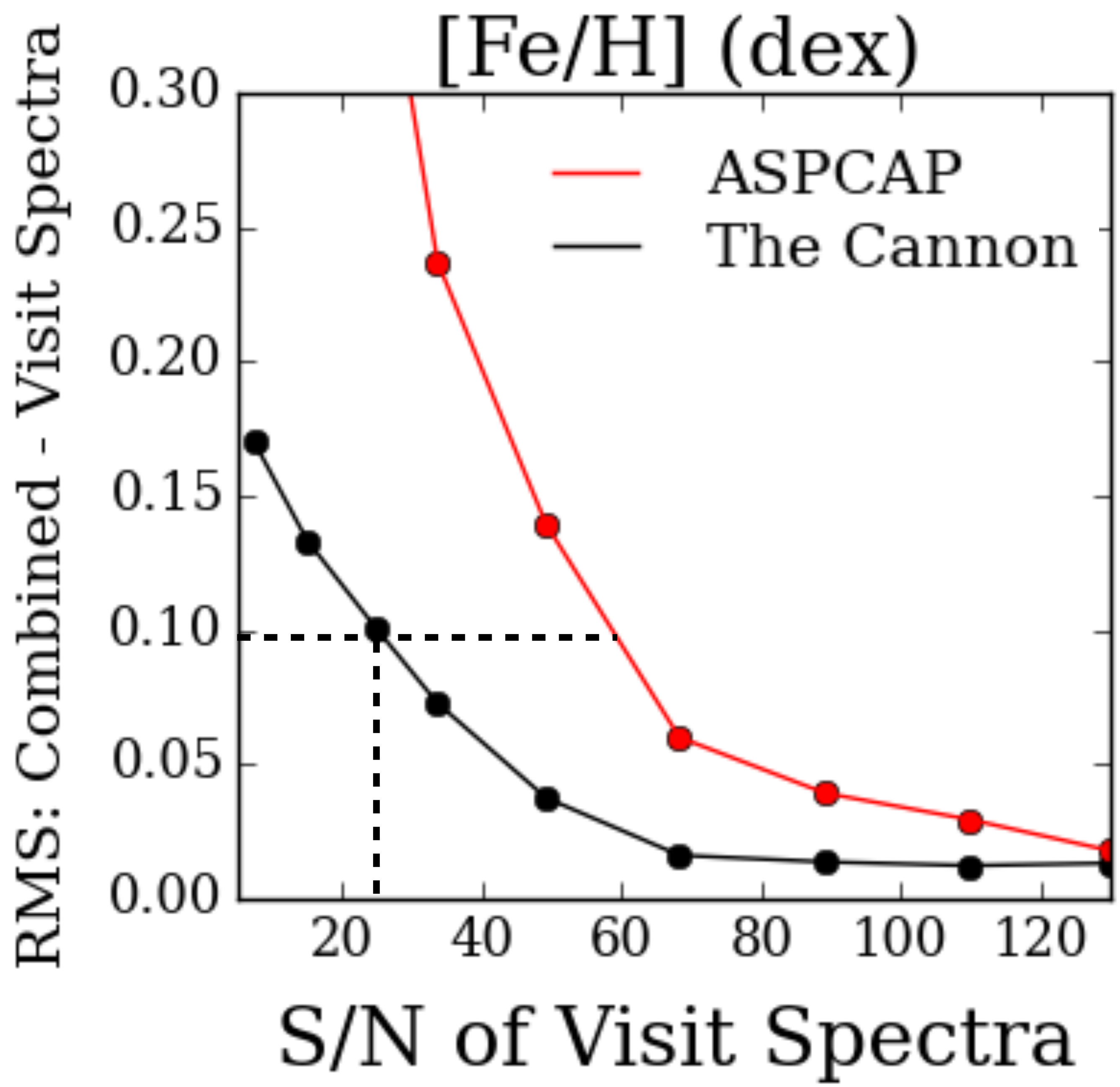
using multiple observations to assess precision

The Cannon delivers 2-3 times smaller errors



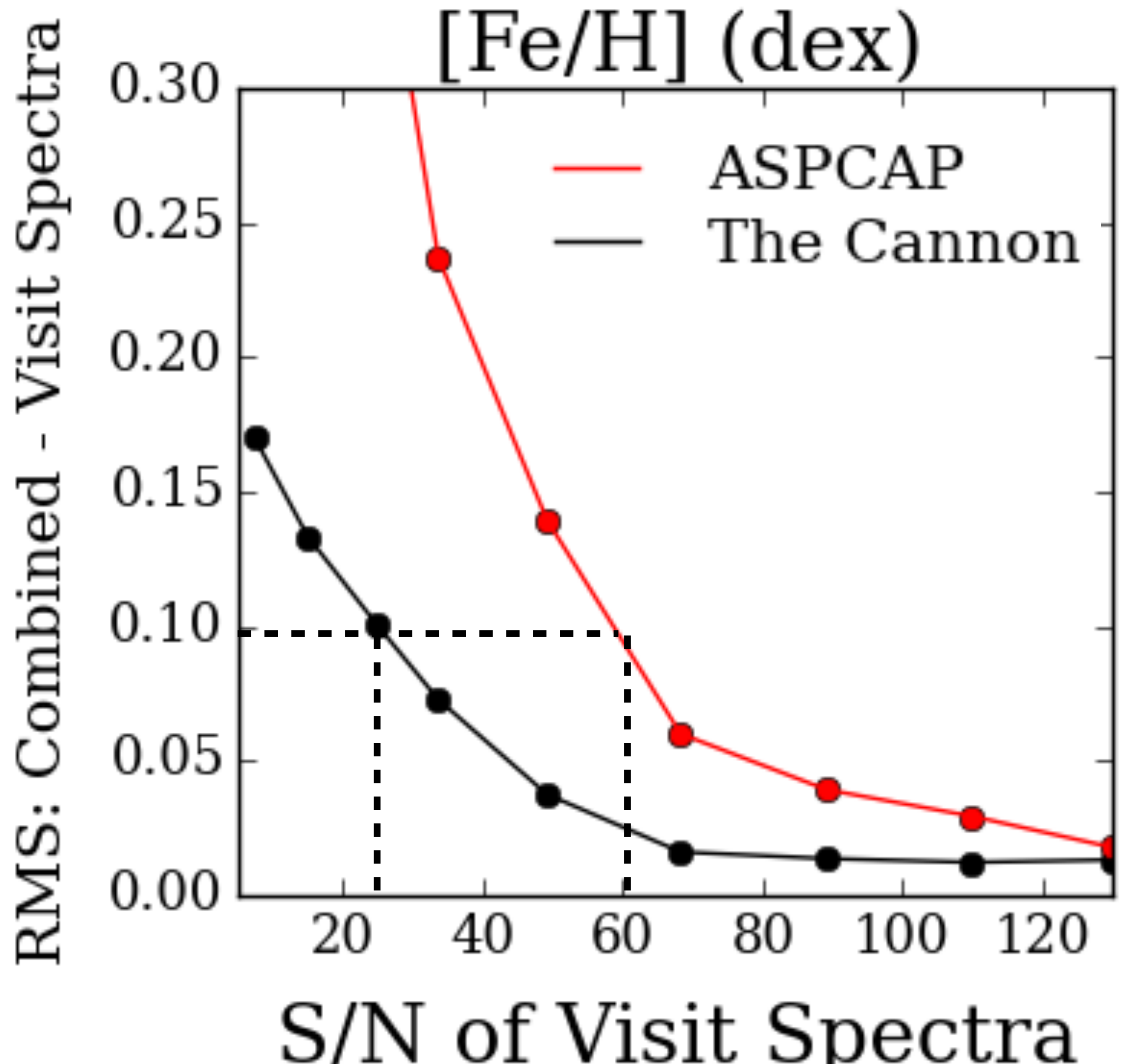
using multiple observations to assess precision

The Cannon delivers 2-3 times smaller errors



using multiple observations to assess precision

The Cannon delivers 2-3 times smaller errors



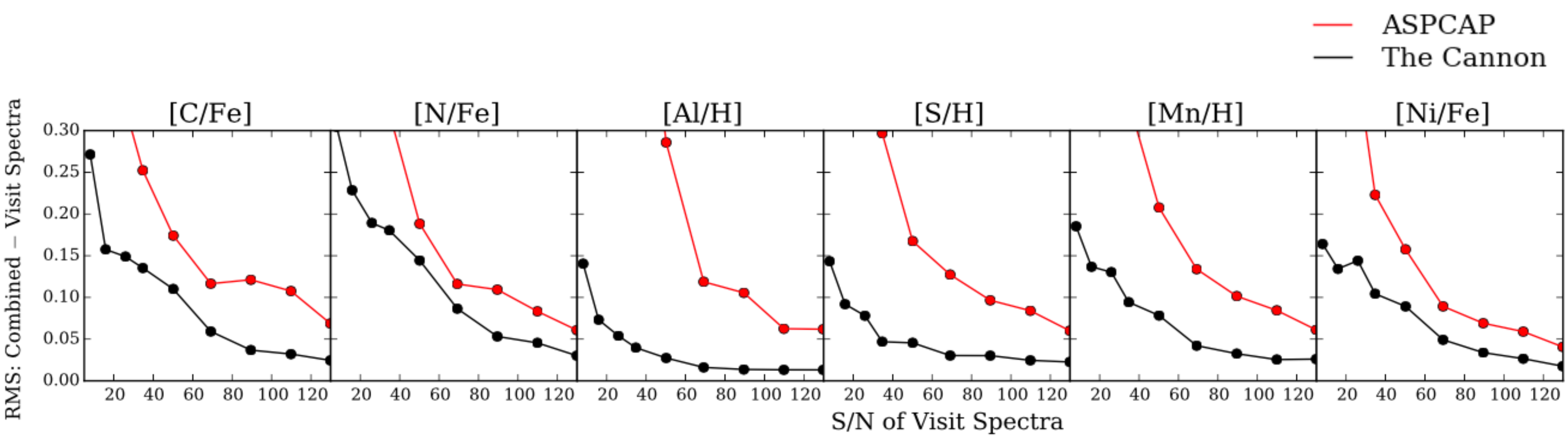
using multiple observations to assess precision

Individual abundances

Add more labels to The Cannon - [X/Fe]

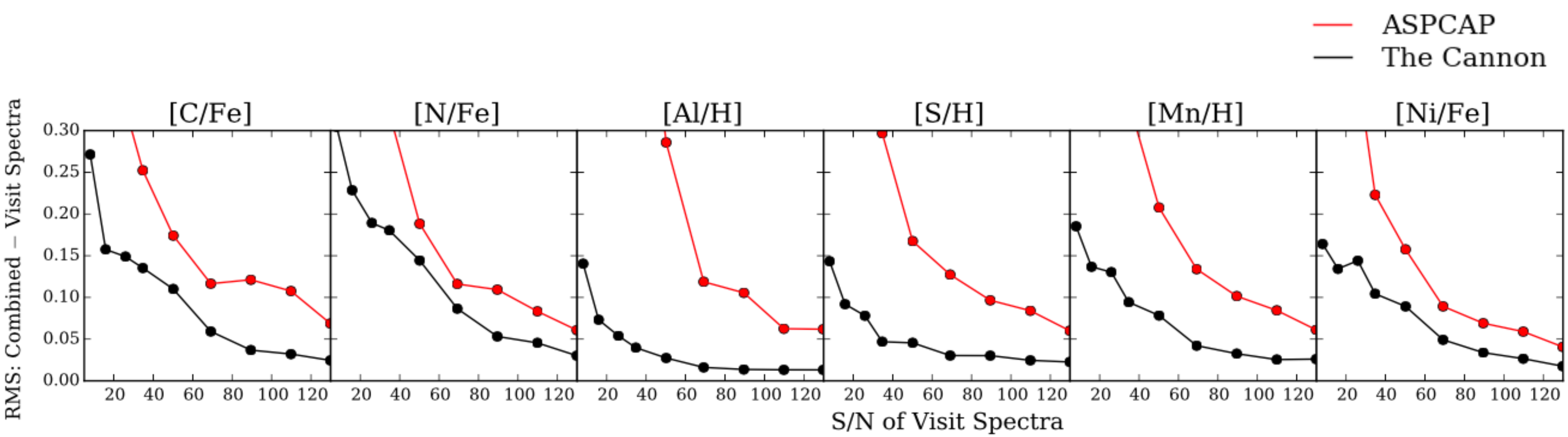
Individual abundances

Add more labels to The Cannon - [X/Fe]



Individual abundances

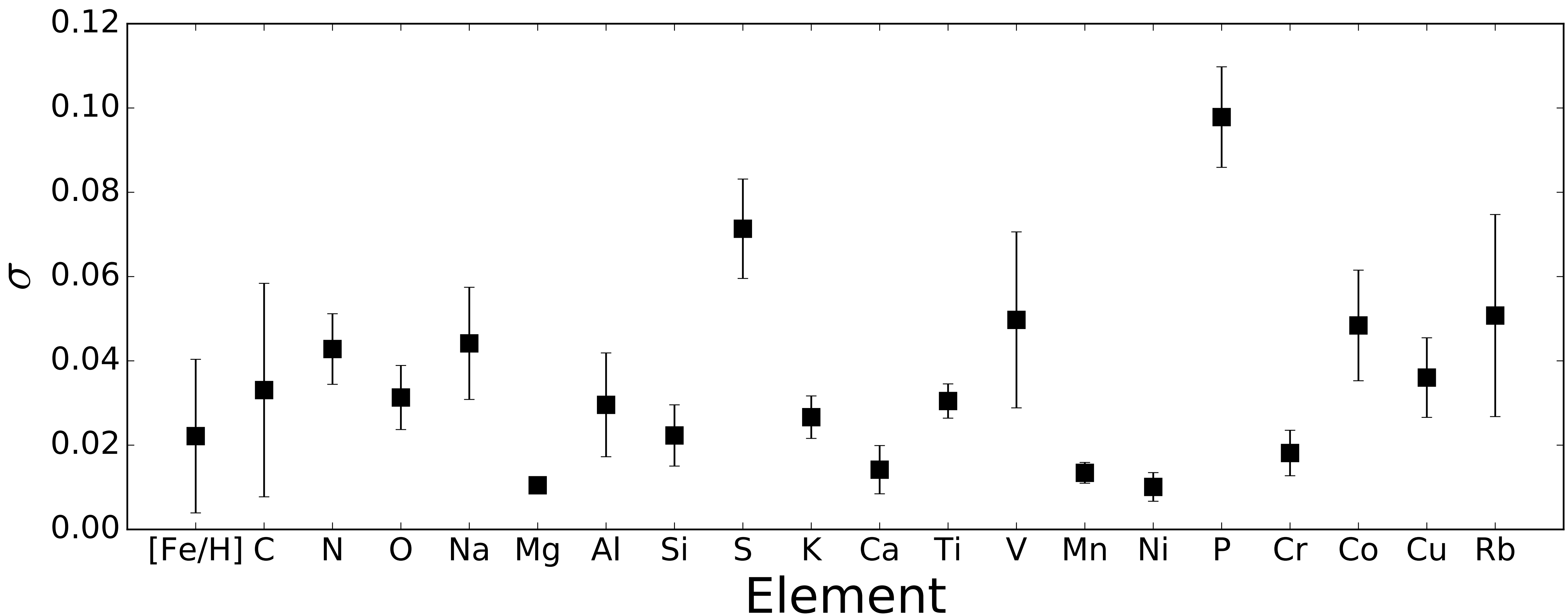
Add more labels to The Cannon - [X/Fe]



APOGEE DR14 data release has catalogue using The Cannon

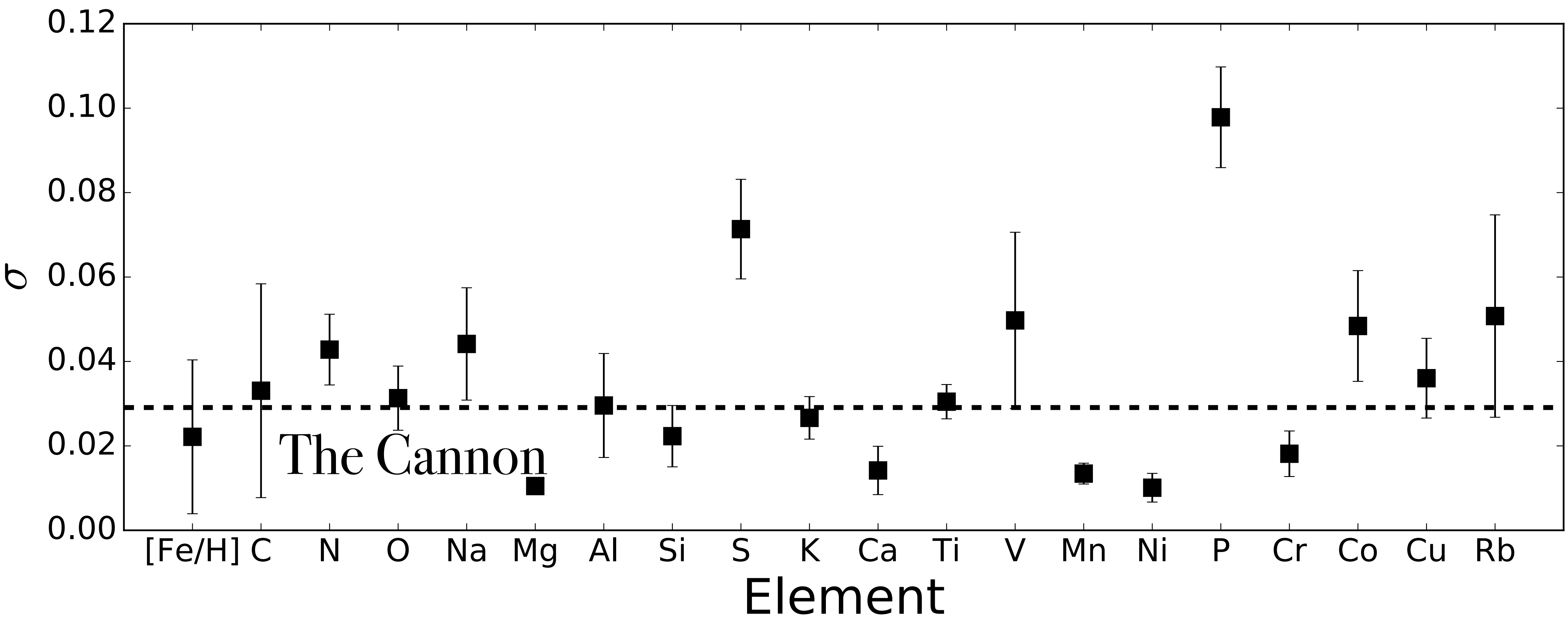
Median precision of 0.03 dex for 20 elements

Carefully assessed precision (Ness et al., 2018)



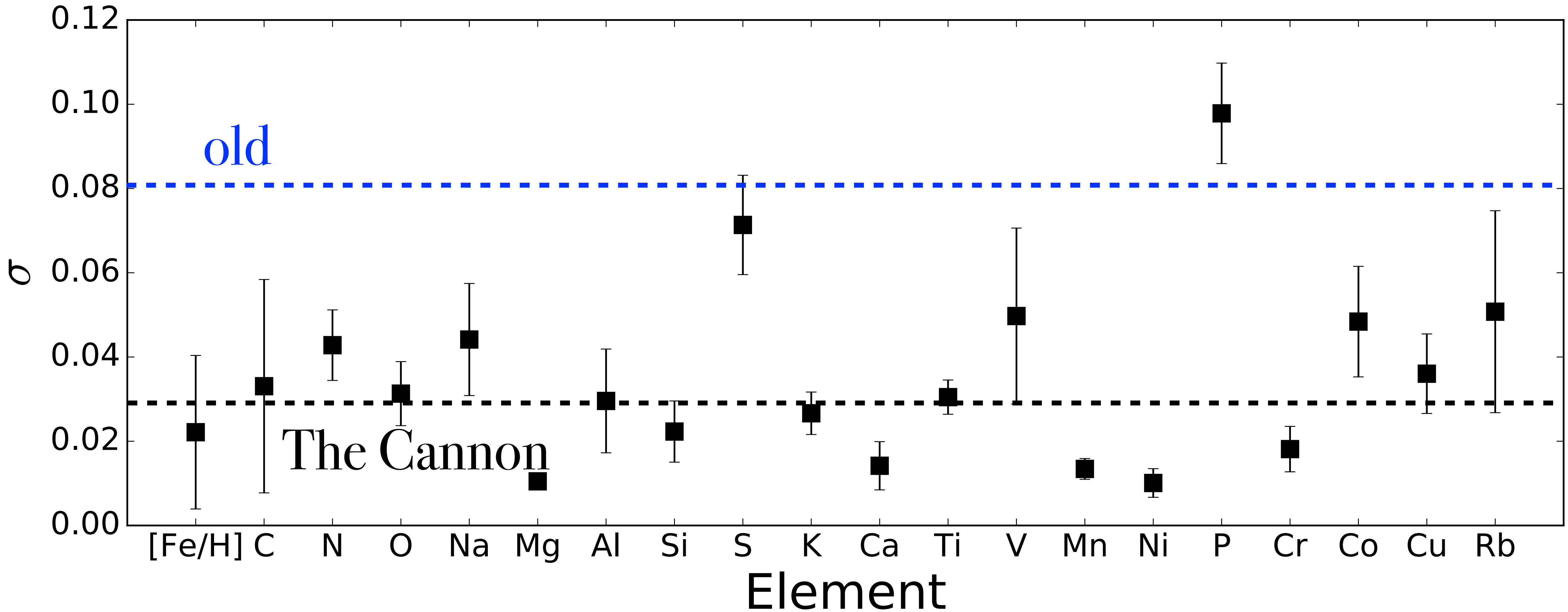
Median precision of 0.03 dex for 20 elements

Carefully assessed precision (Ness et al., 2018)



Median precision of 0.03 dex for 20 elements

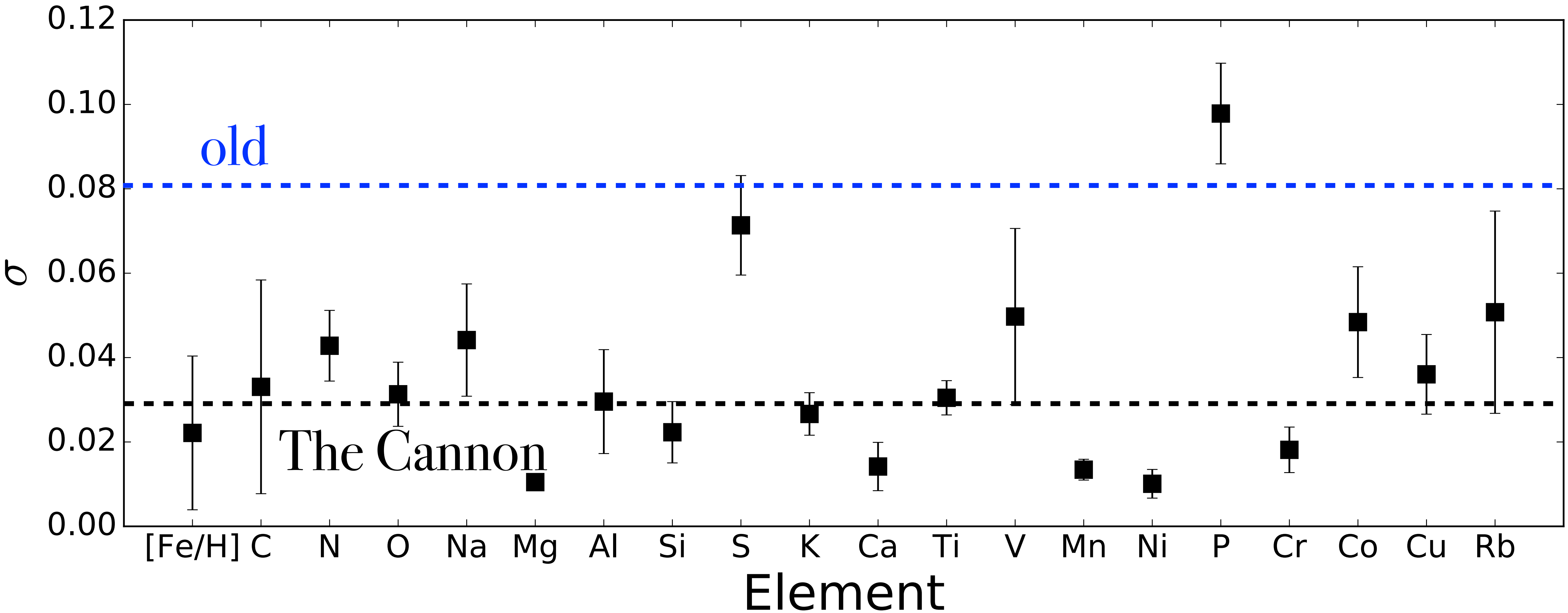
Carefully assessed precision (Ness et al., 2018)



Median precision of 0.03 dex for 20 elements

Carefully assessed precision (Ness et al., 2018)

with higher precision, more discriminating power in measurements



Galah


Sven Buder

Galah

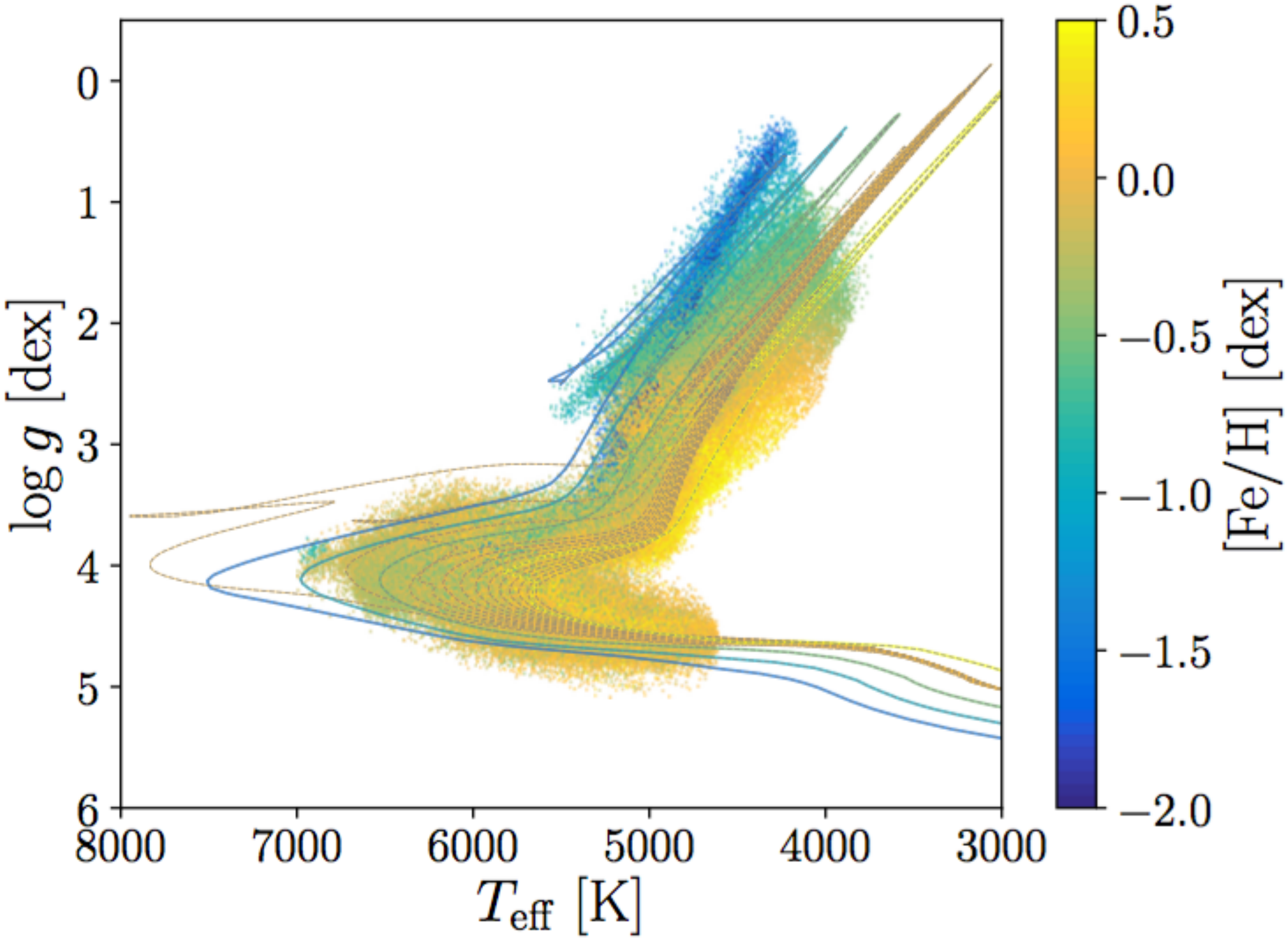
Sven Buder

350,000 stars (R=30,000) 23 element abundances inc. Ti, Si, Mg, O, Li, Ba, Y

Galah


Sven Buder

350,000 stars (R=30,000) 23 element abundances inc. Ti, Si, Mg, O, Li, Ba, Y



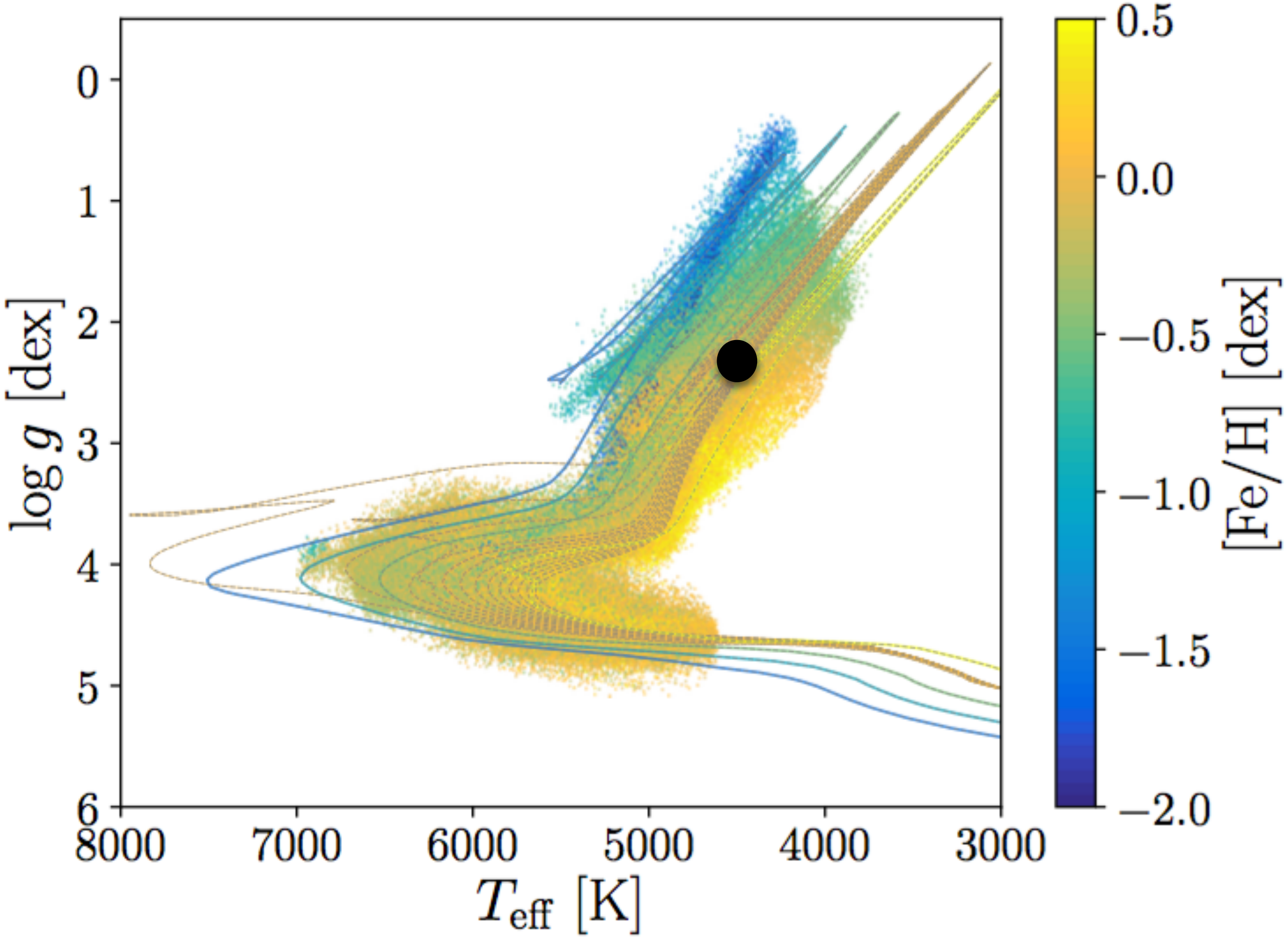
Gaia DR2 spectroscopic complement

(Buder et al., 2018a,b)

Galah

Sven Buder

350,000 stars (R=30,000) 23 element abundances inc. Ti, Si, Mg, O, Li, Ba, Y



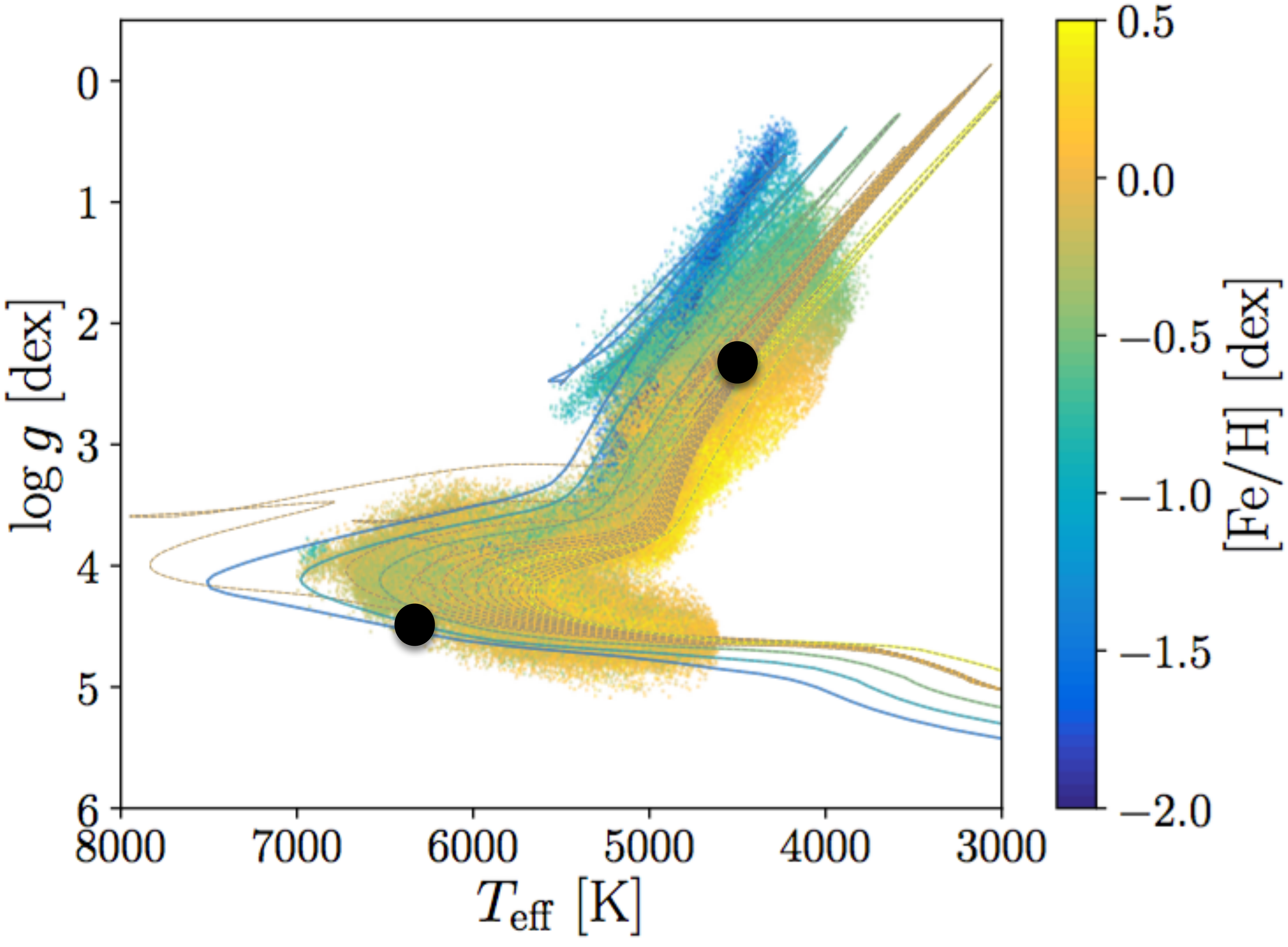
Gaia DR2 spectroscopic complement

(Buder et al., 2018a,b)

Galah

Sven Buder

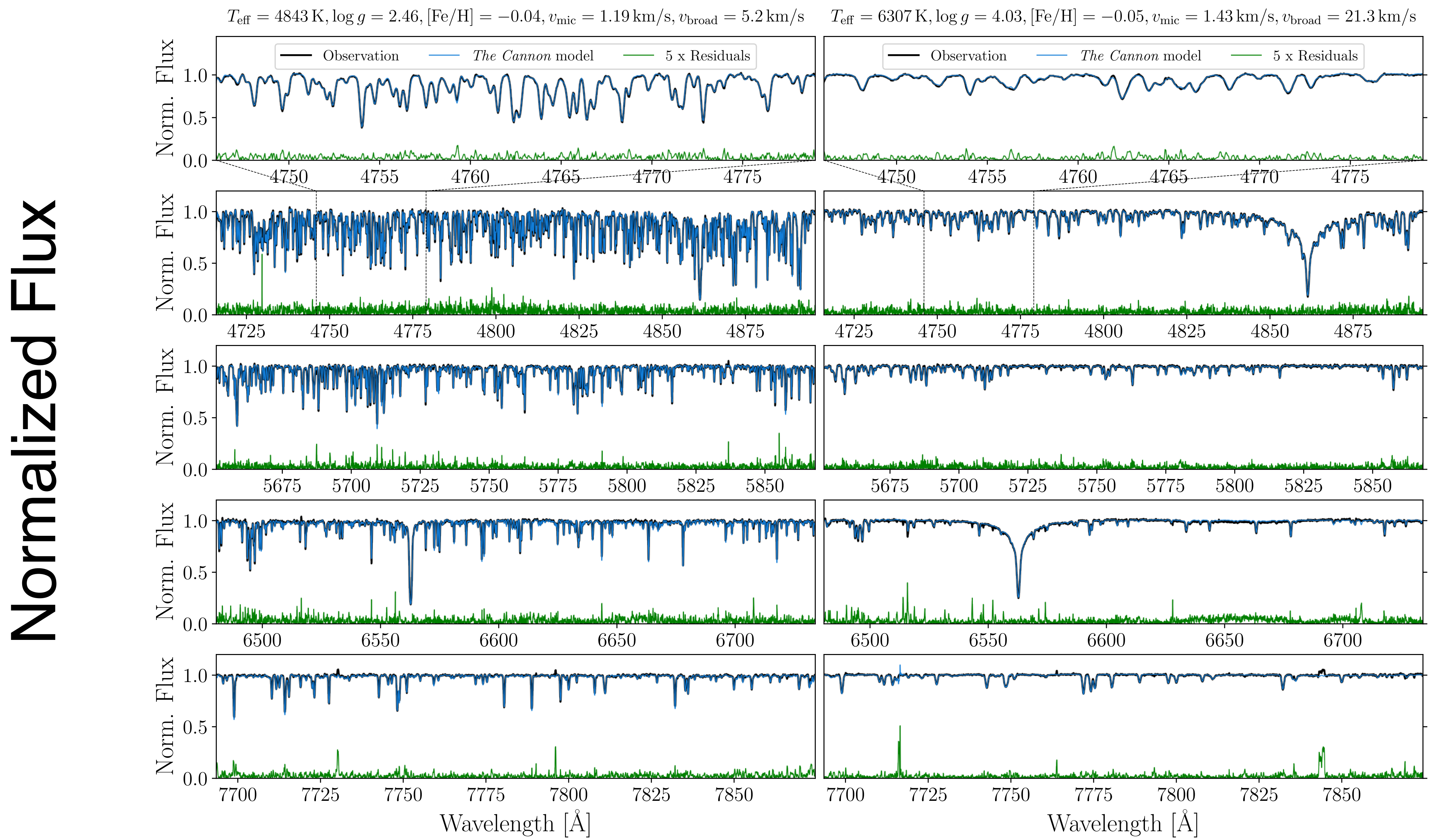
350,000 stars (R=30,000) 23 element abundances inc. Ti, Si, Mg, O, Li, Ba, Y



Gaia DR2 spectroscopic complement

(Buder et al., 2018a,b)

Galah model versus data - 23 elements

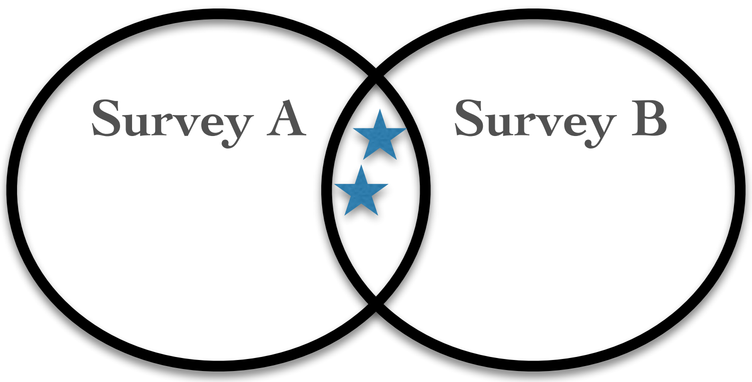


Applications…

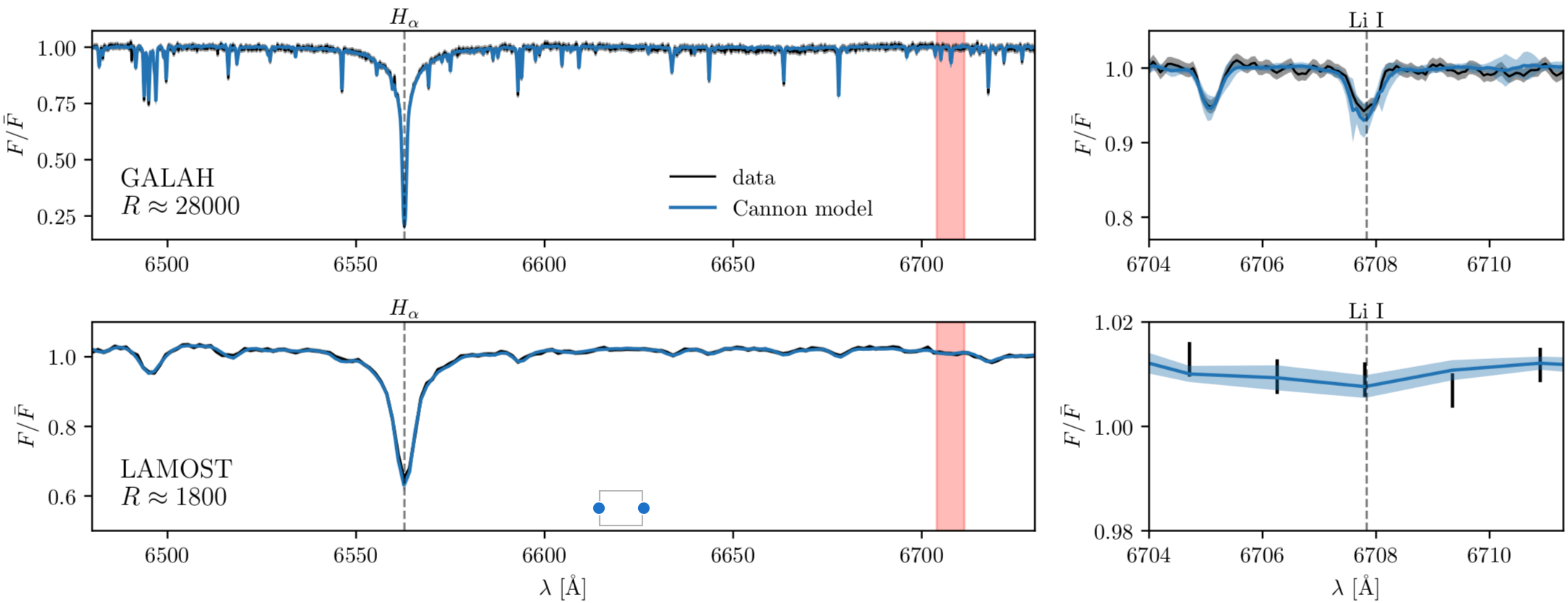
Low-resolution spectra

Individual abundances from LAMOST

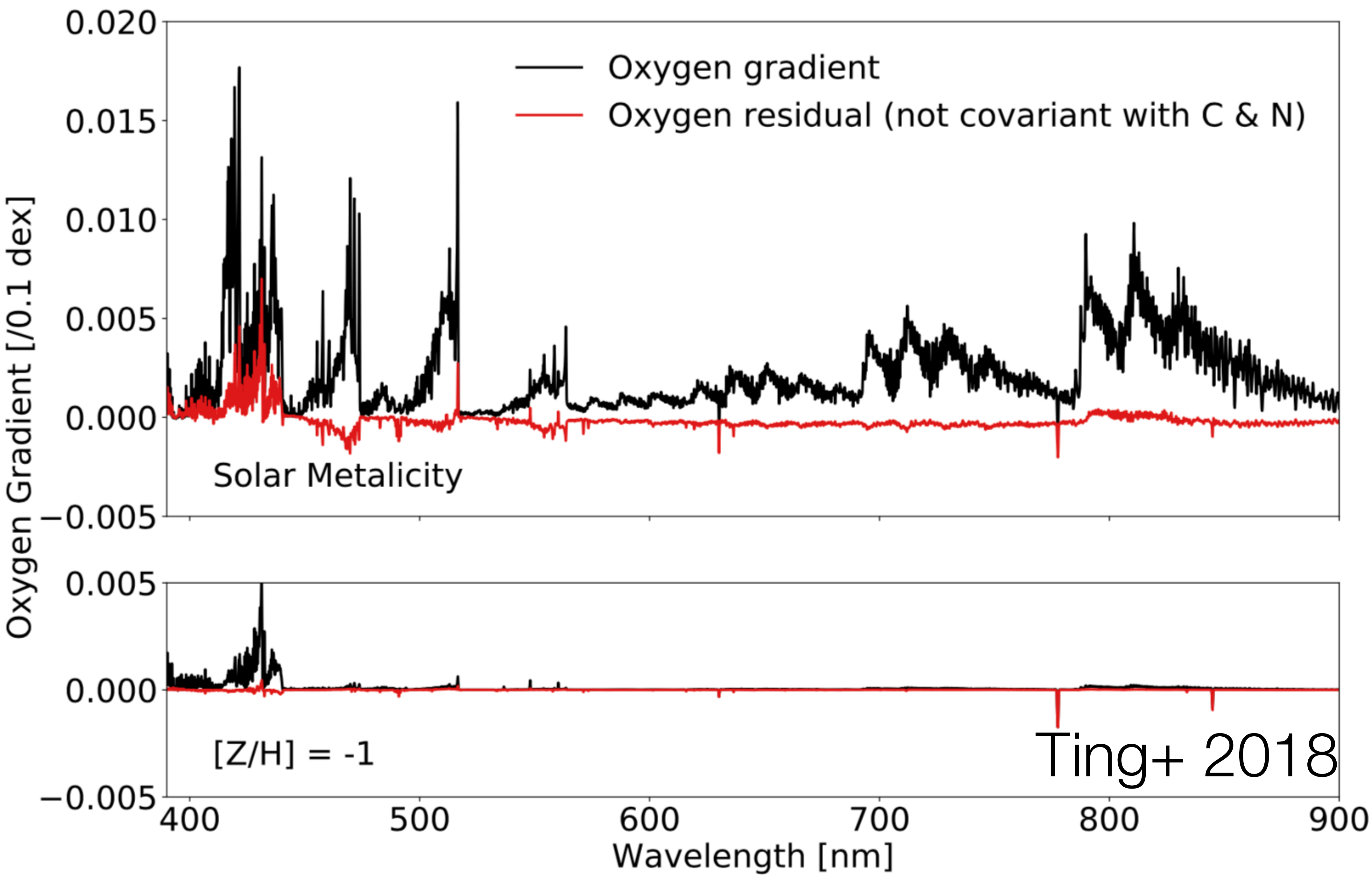
Wheeler & Ness (in prep) - catalogue of 4 million stars with 7 abundances



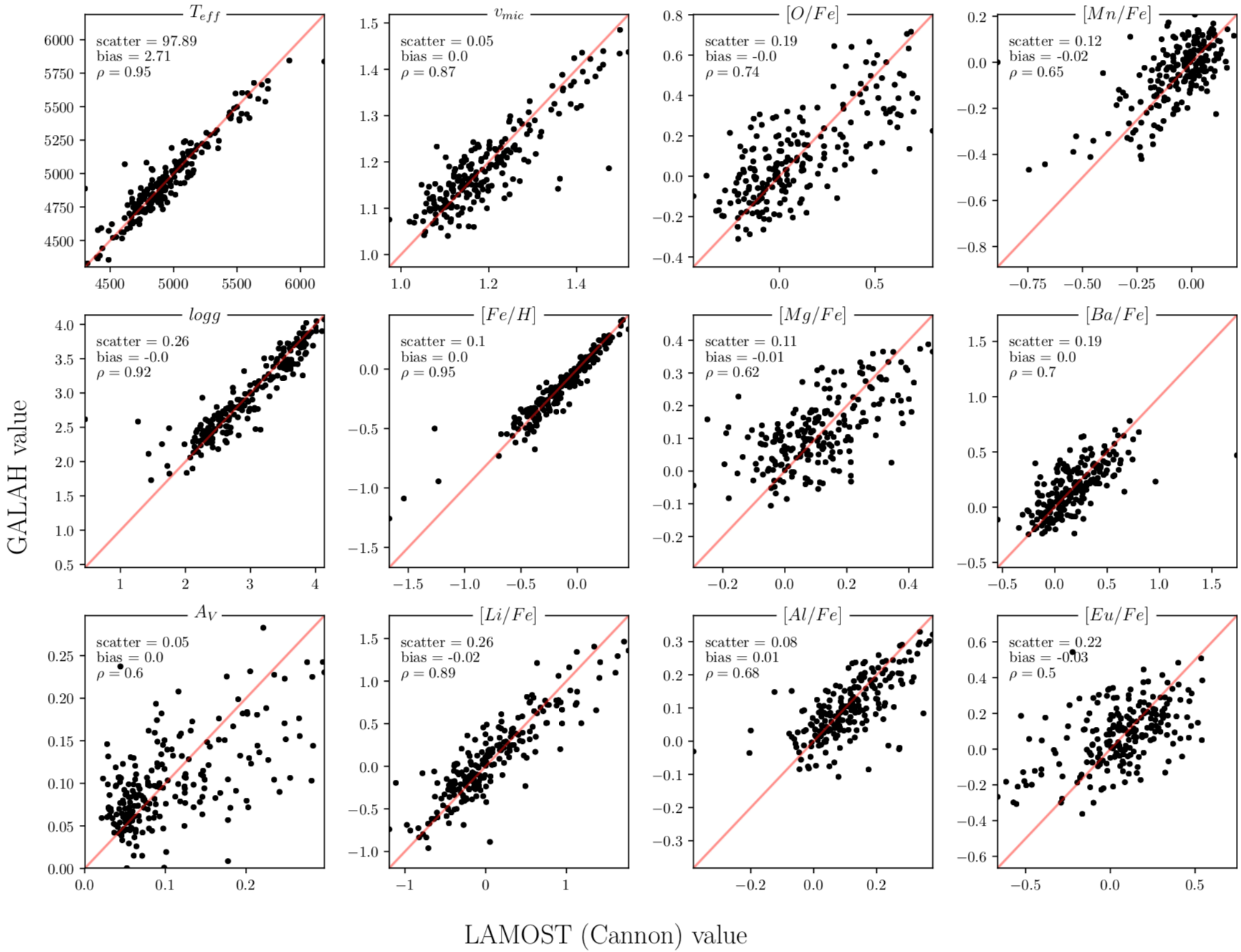
labels from A (GALAH)
spectra from B (LAMOST)



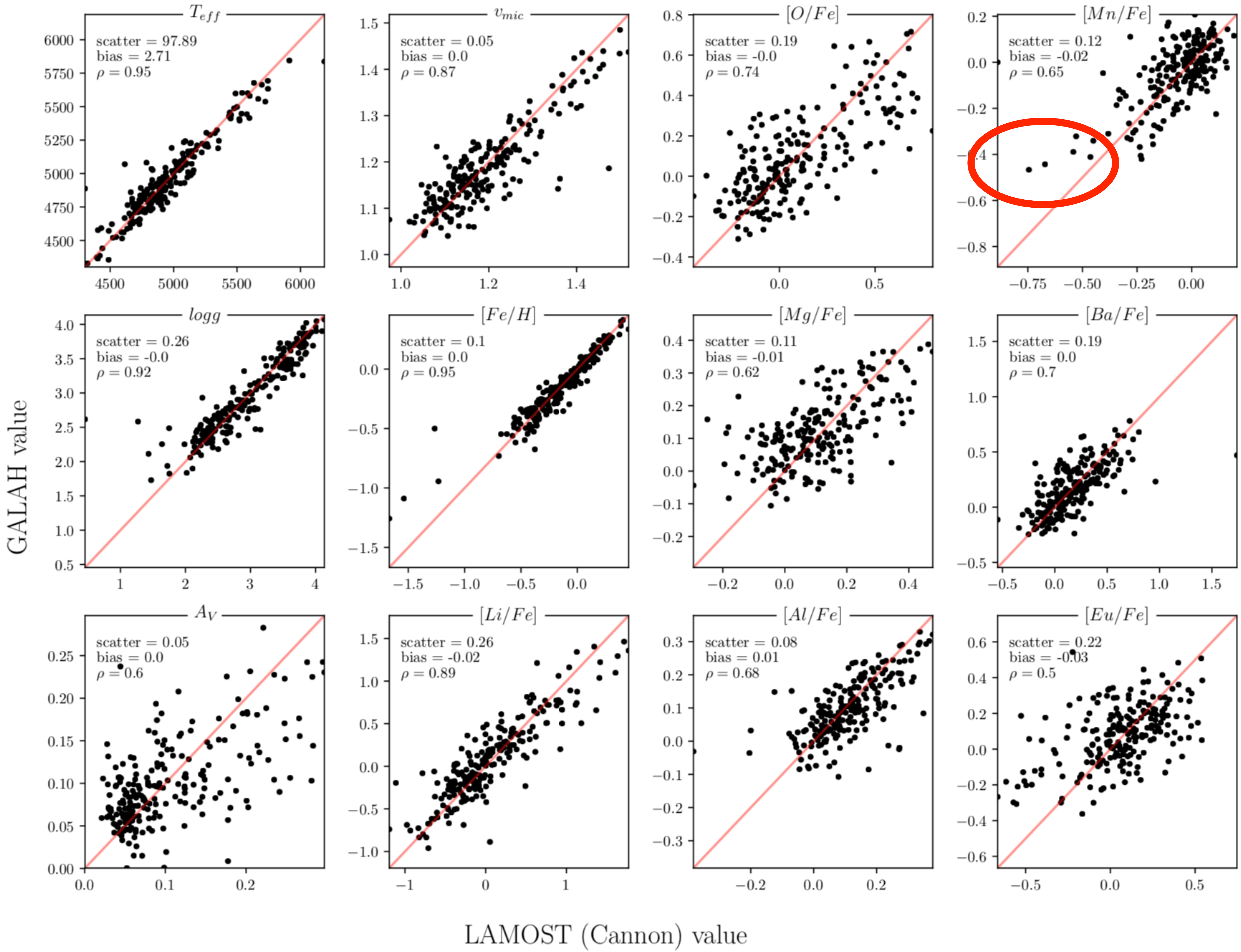
You don’t always need nice lines



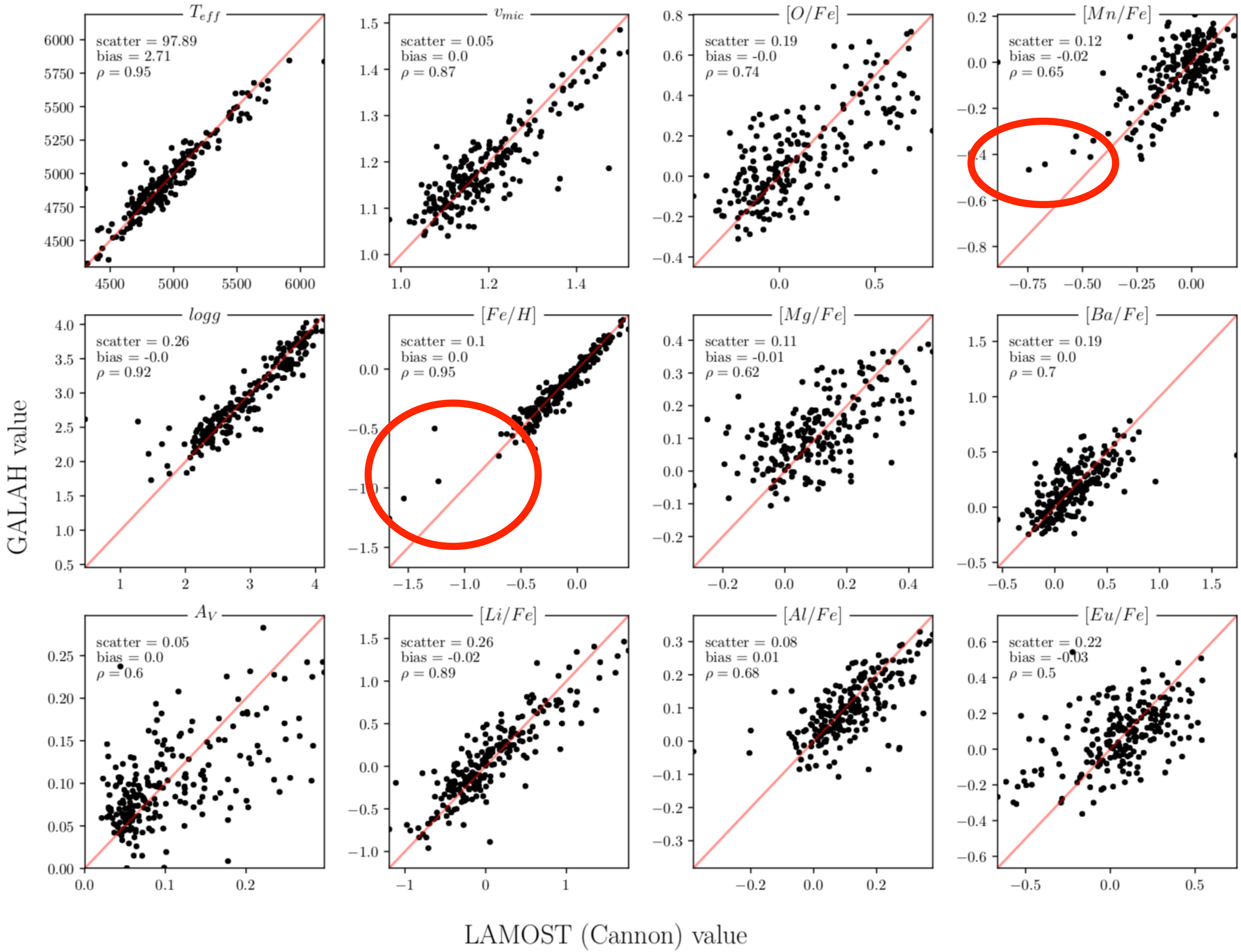
Cross-validation



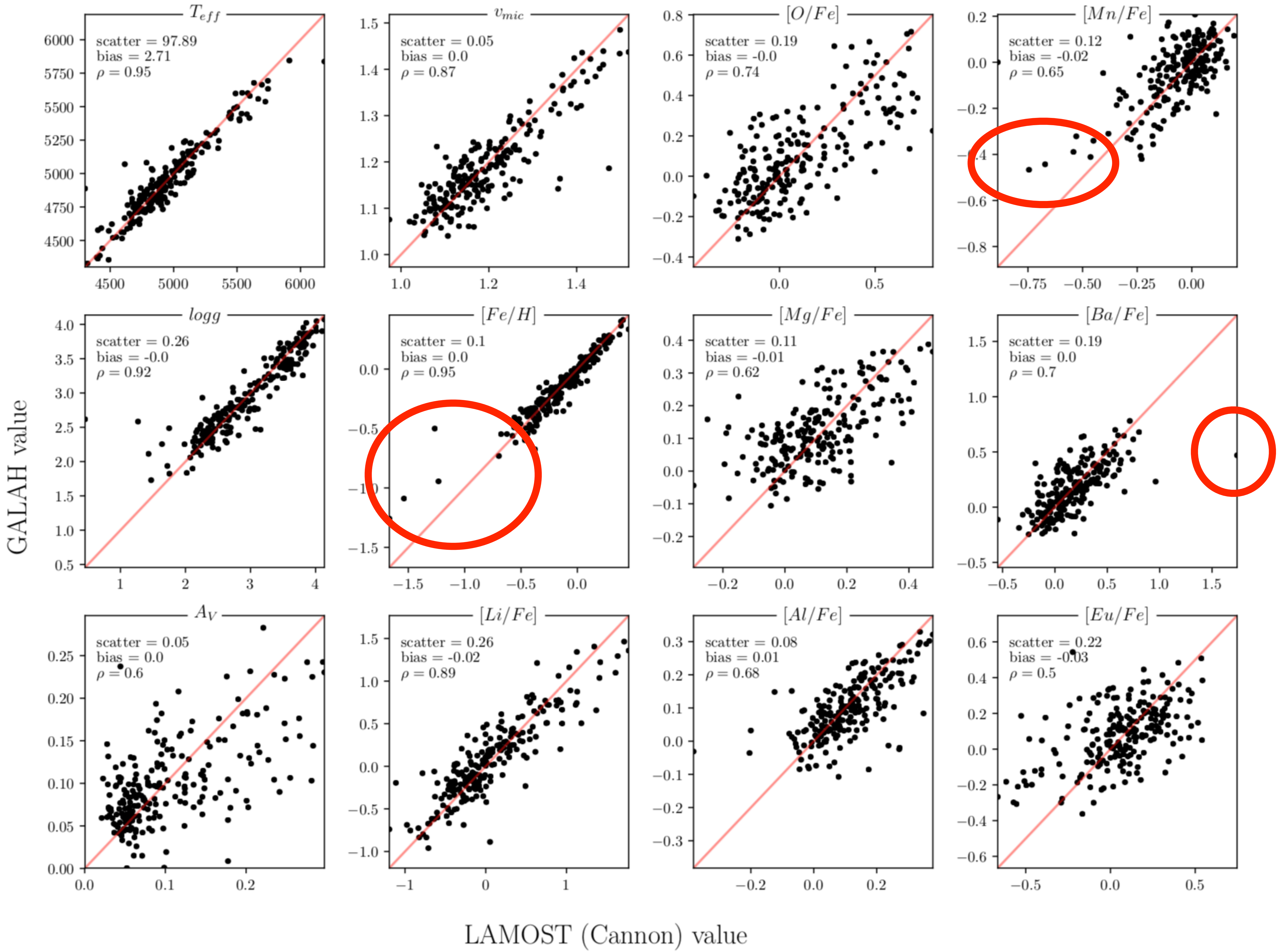
Cross-validation



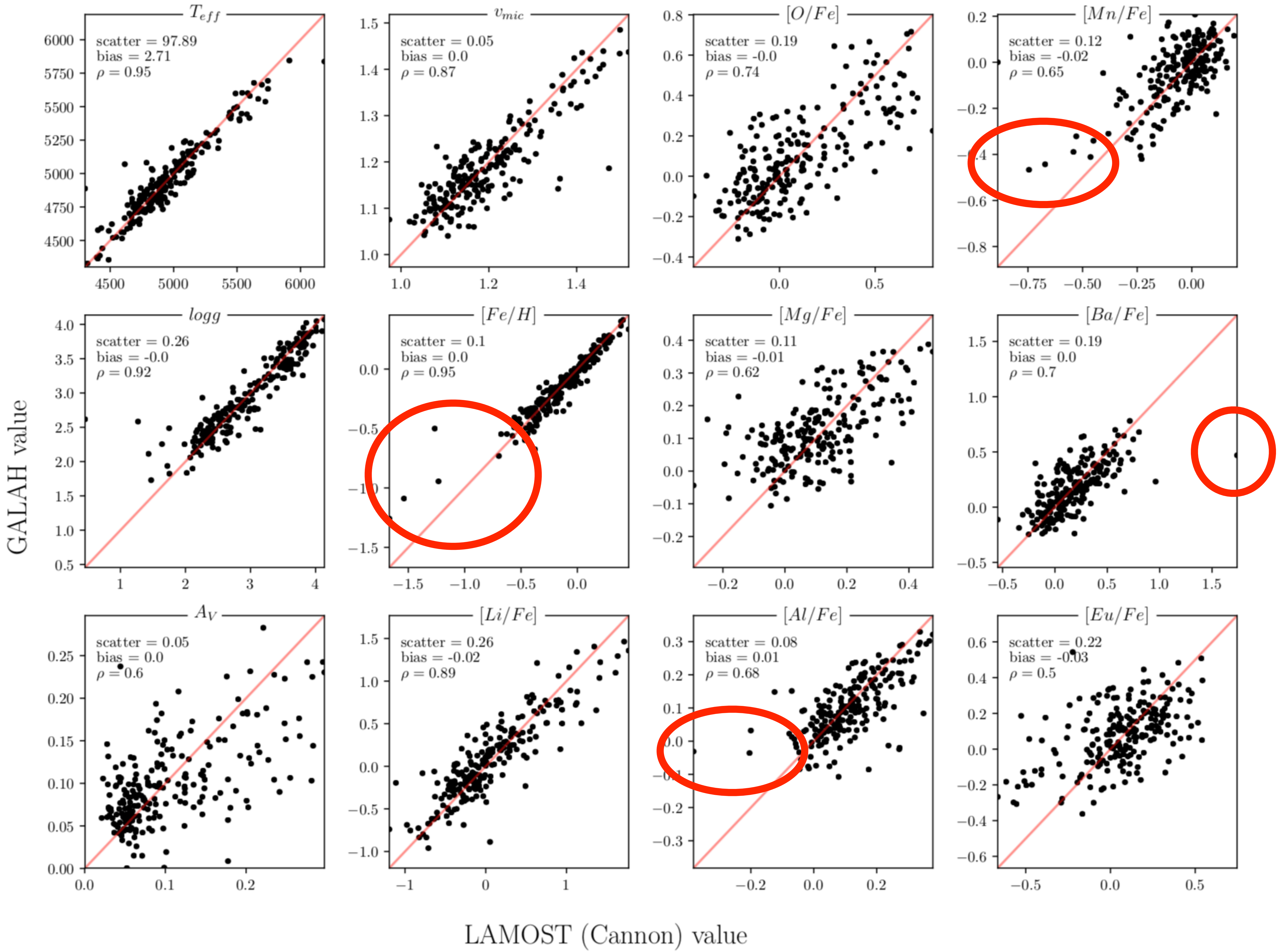
Cross-validation



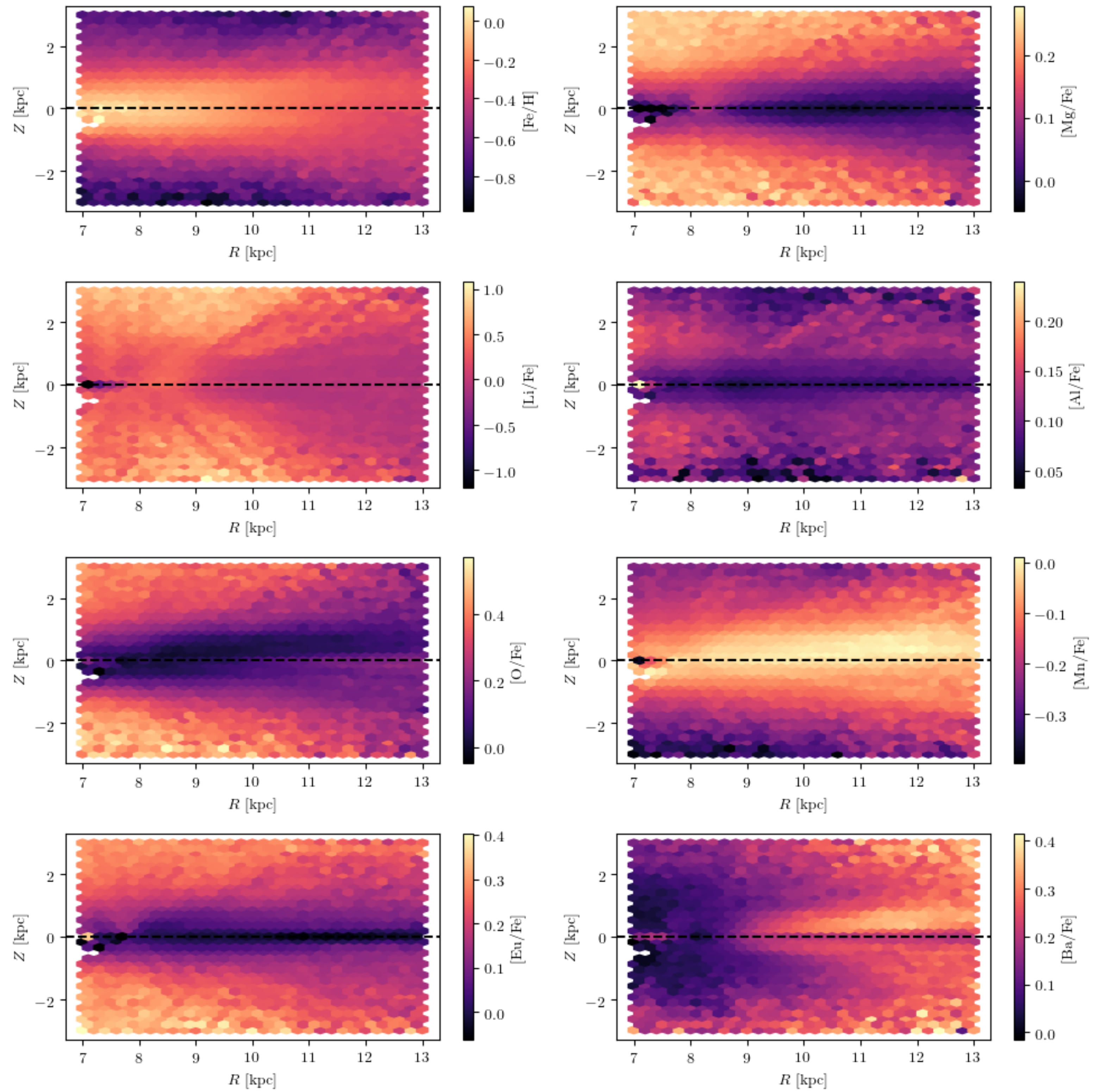
Cross-validation



Cross-validation

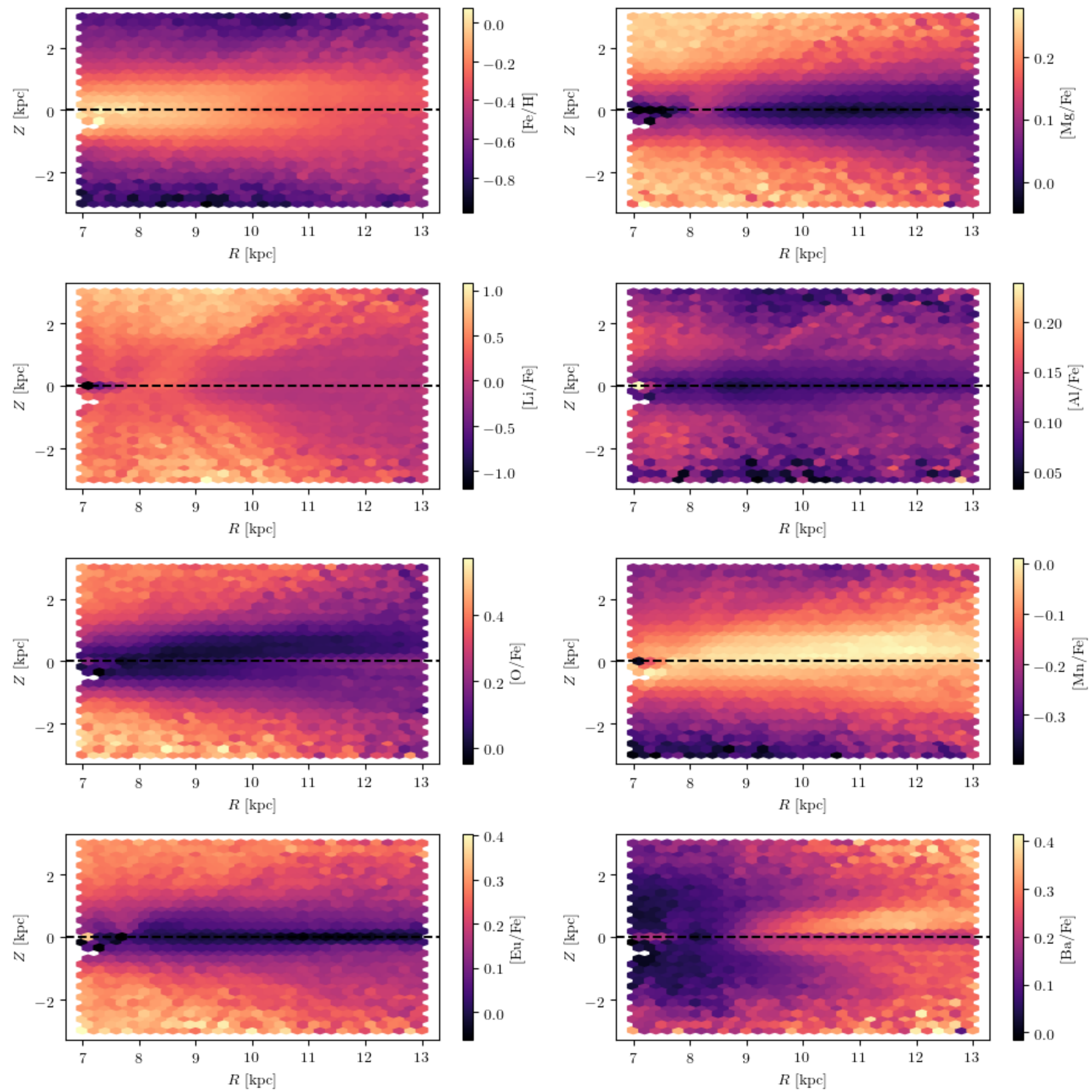


Opportunity - do this for any low resolution spectra (i.e. SLOAN)



Opportunity - do this for any low resolution spectra (i.e. SLOAN)

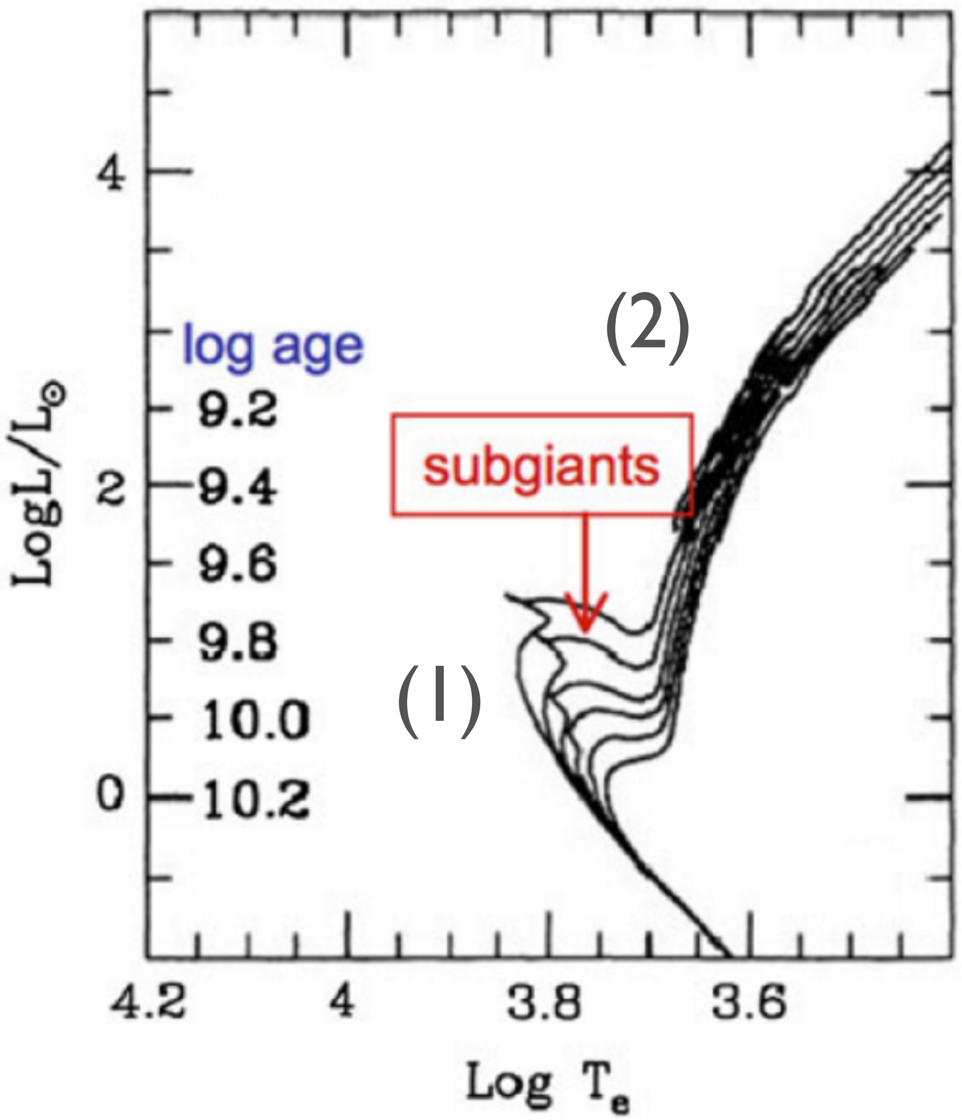
Can learn a lot from many low-precision measurements



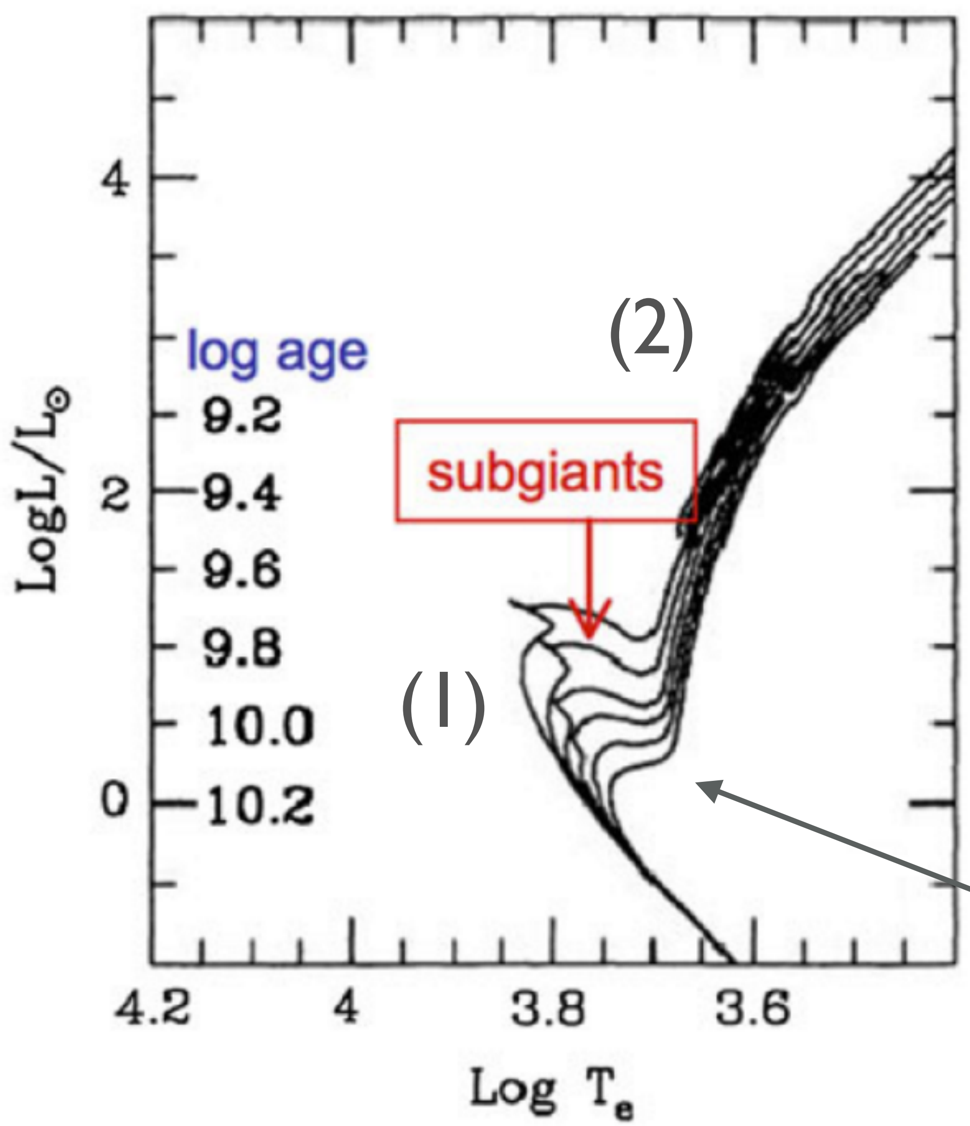
Applications to Galactic archeology…

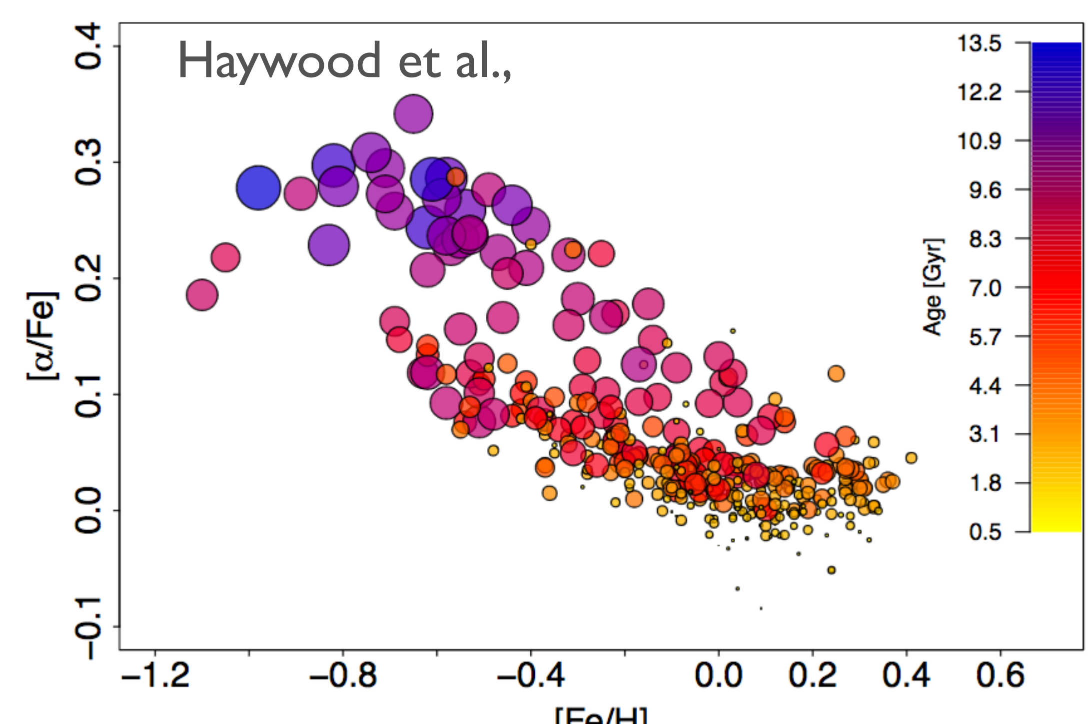
Empirical Spectroscopic ages

How are ages typically measured?

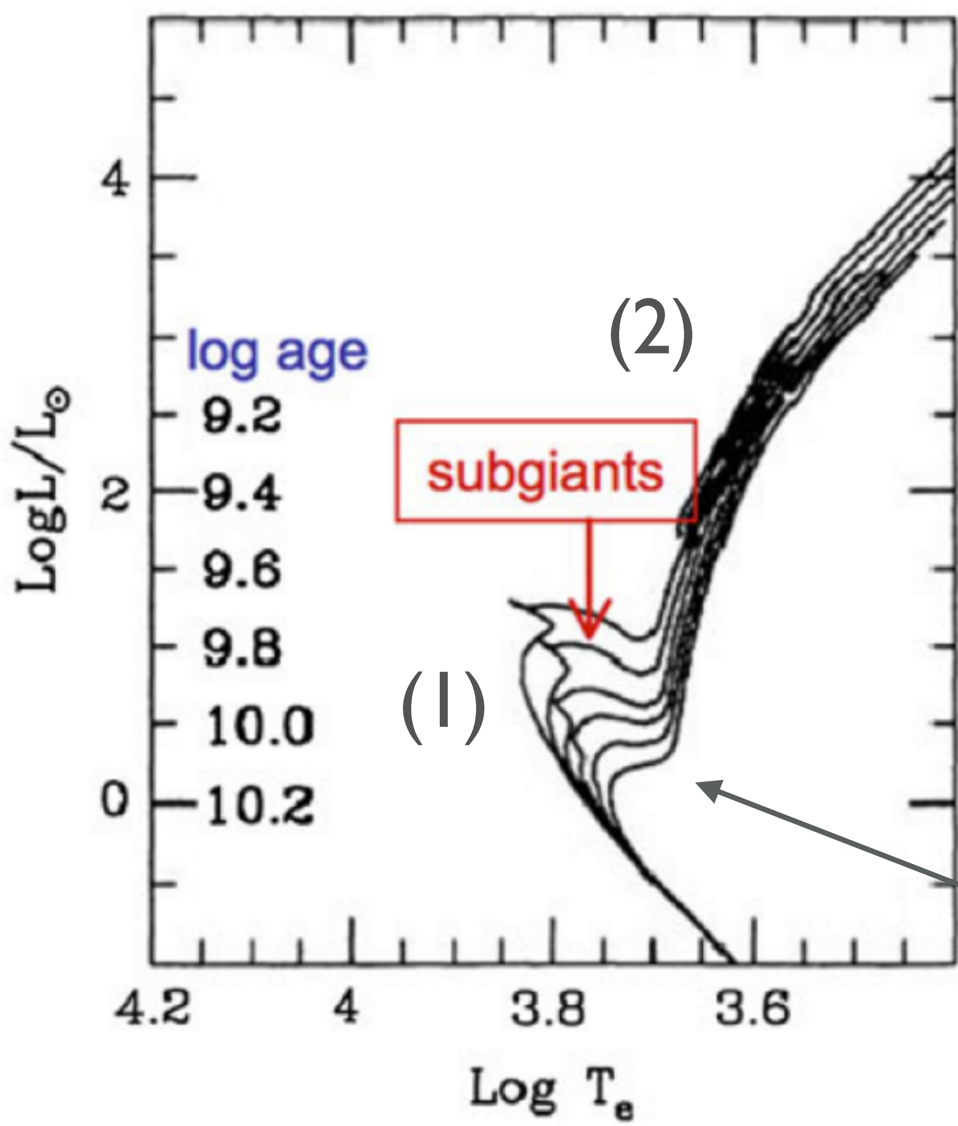


How are ages typically measured?

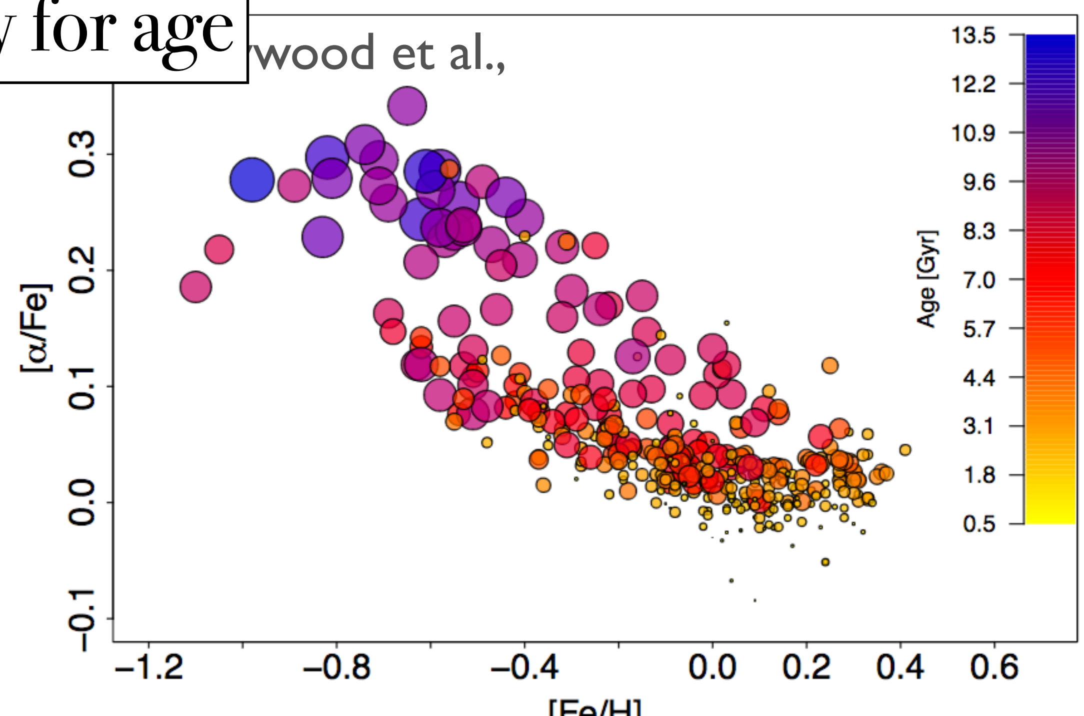




How are ages typically measured?

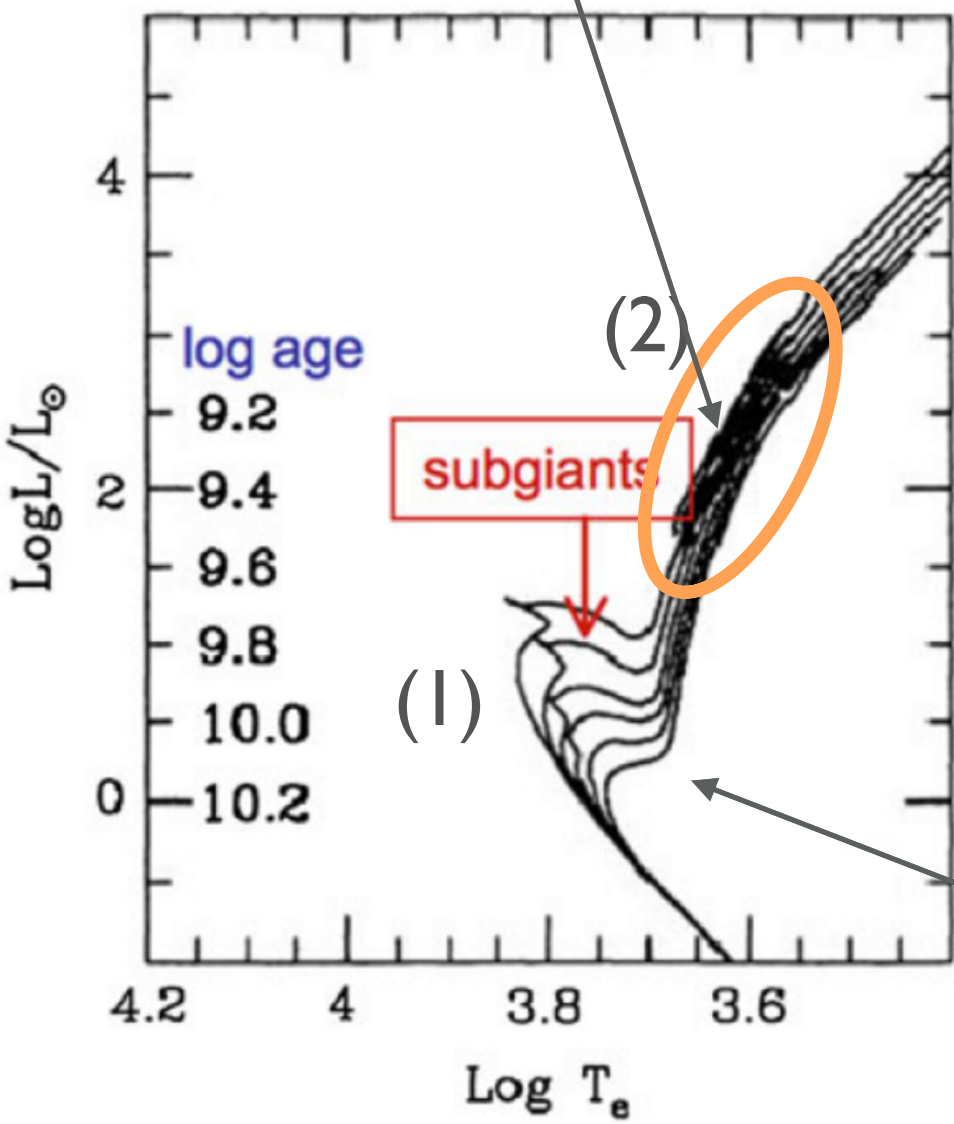


proxy for age

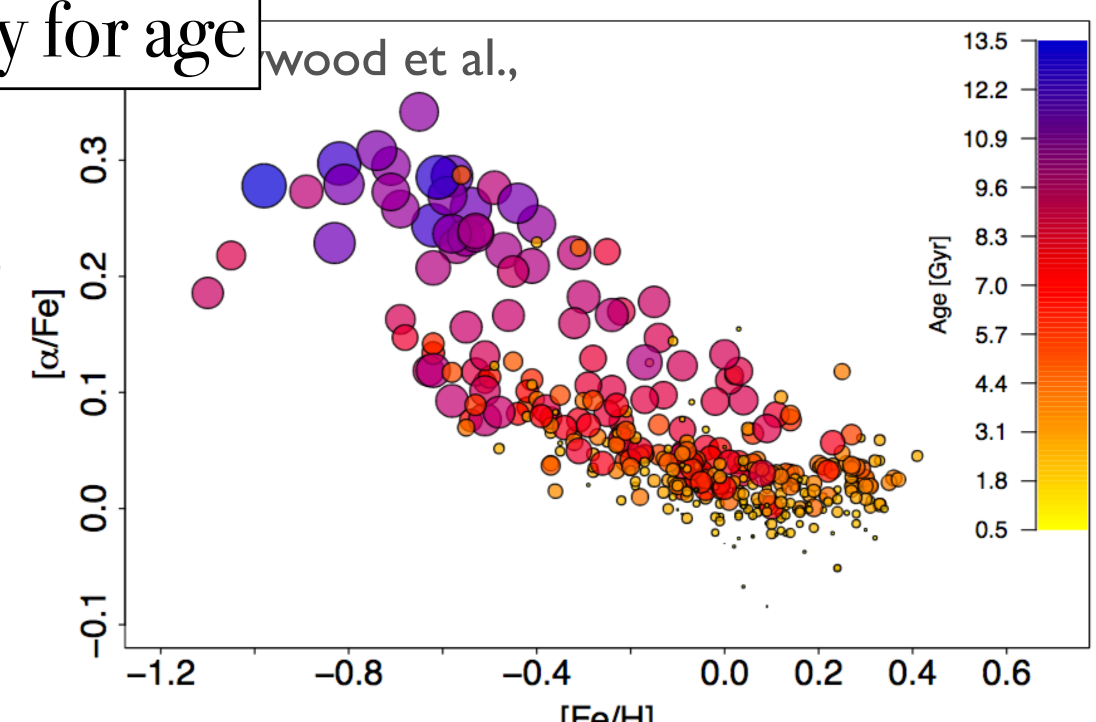


How are ages typically measured?

giants: asteroseismology: Kepler, CoRoT = mass

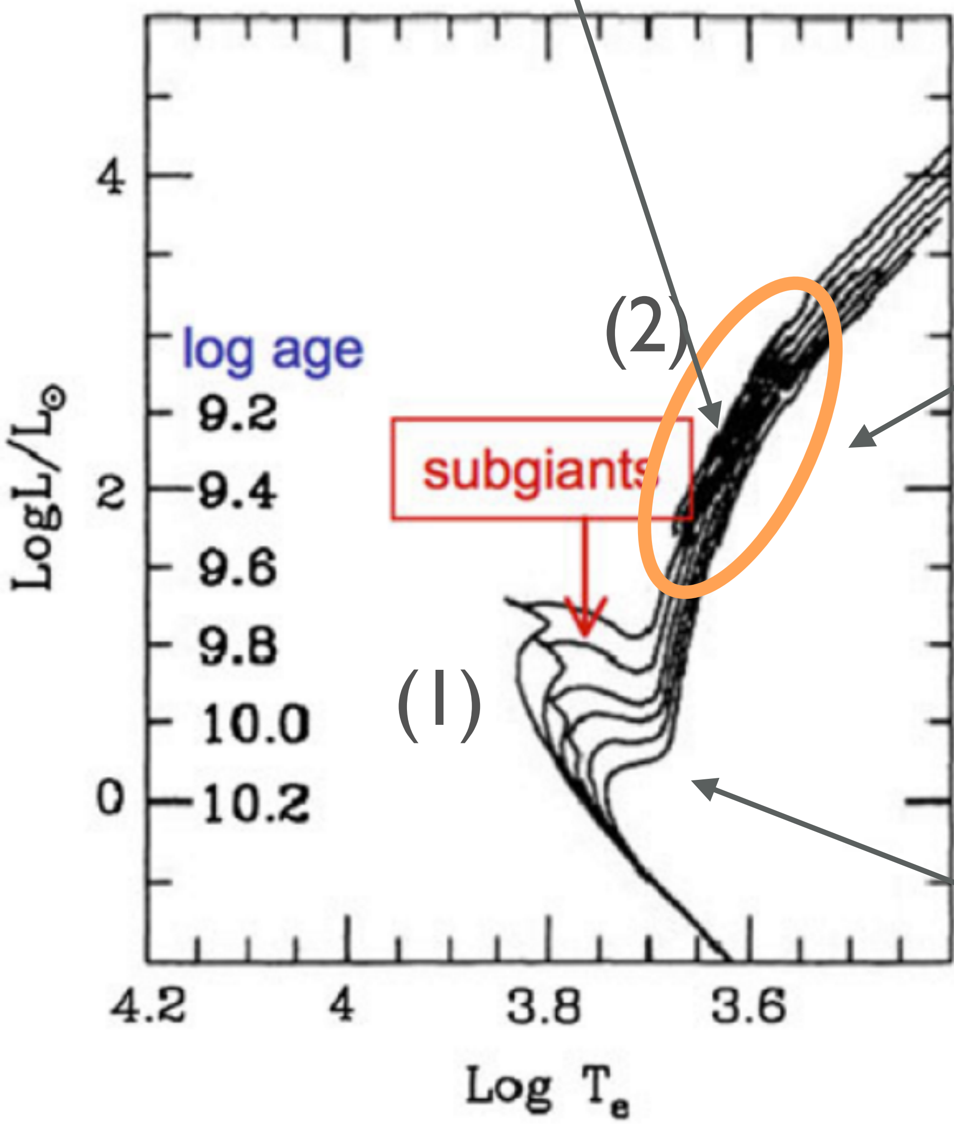


proxy for age

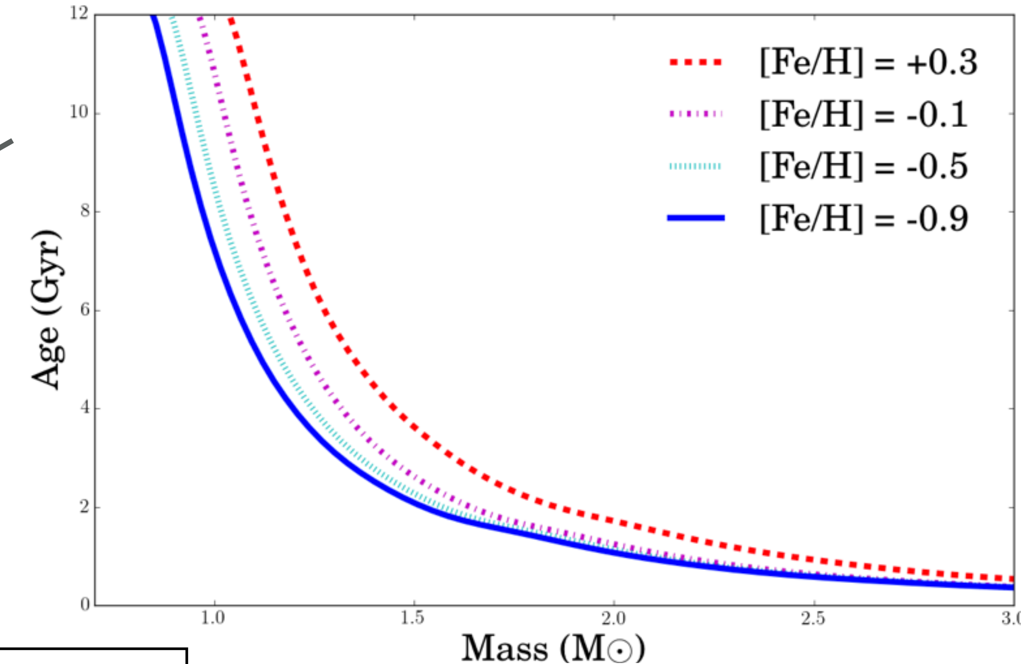


How are ages typically measured?

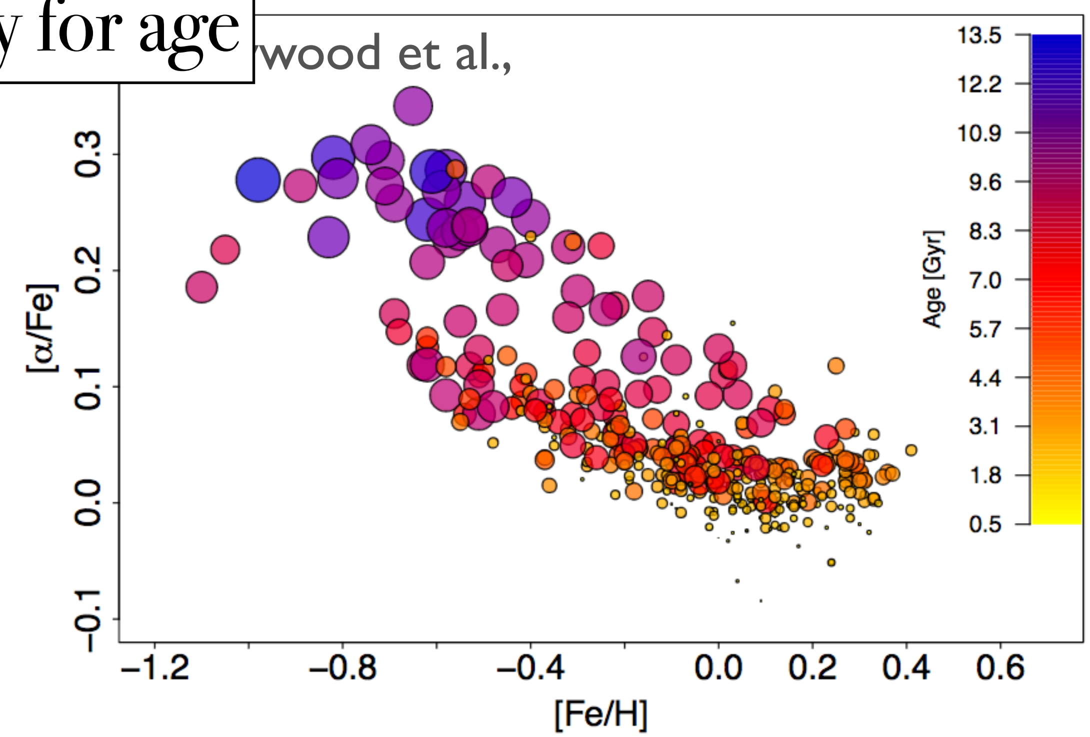
giants: asteroseismology: Kepler, CoRoT = mass



giant masses → giant ages

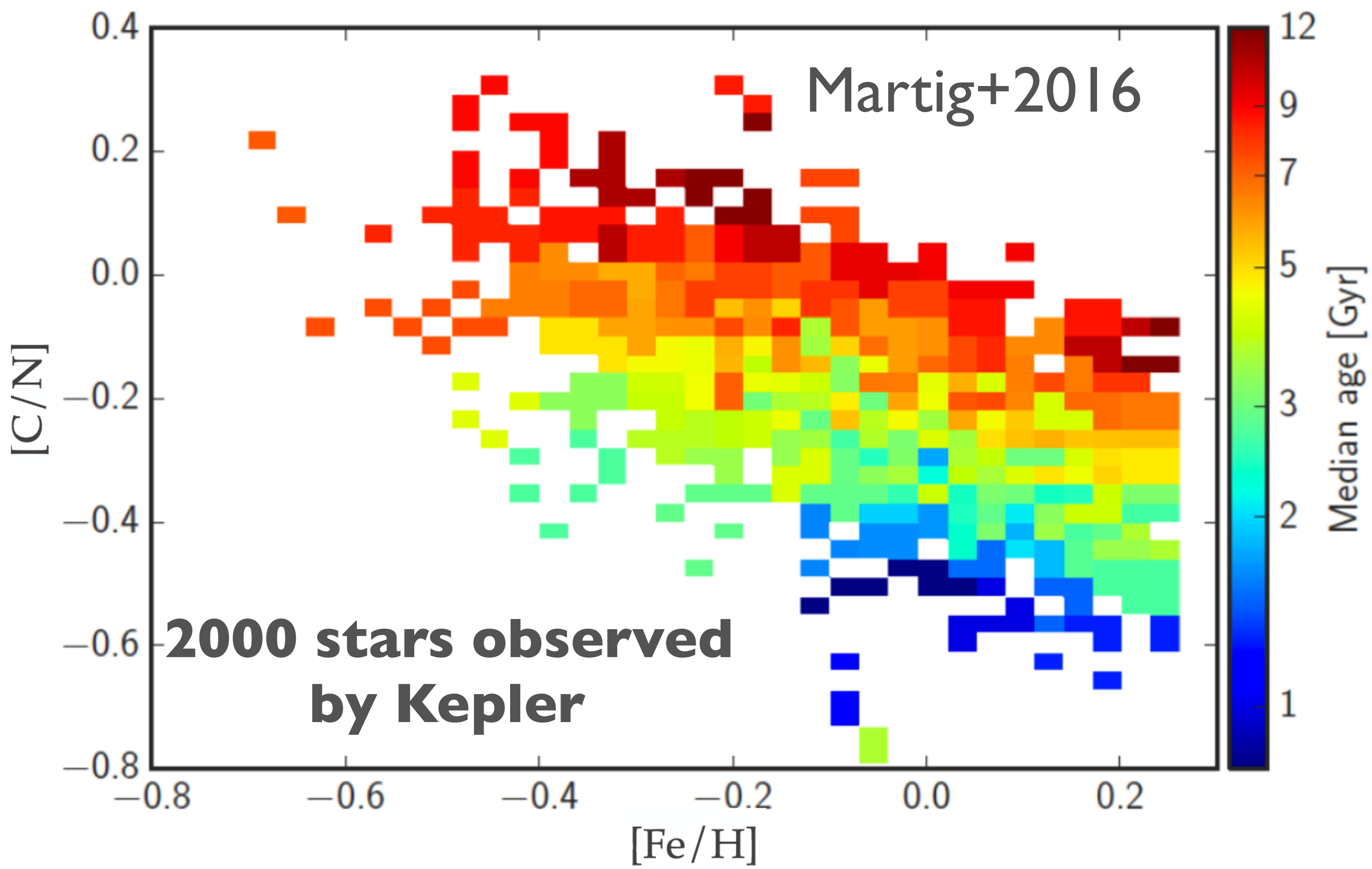


proxy for age



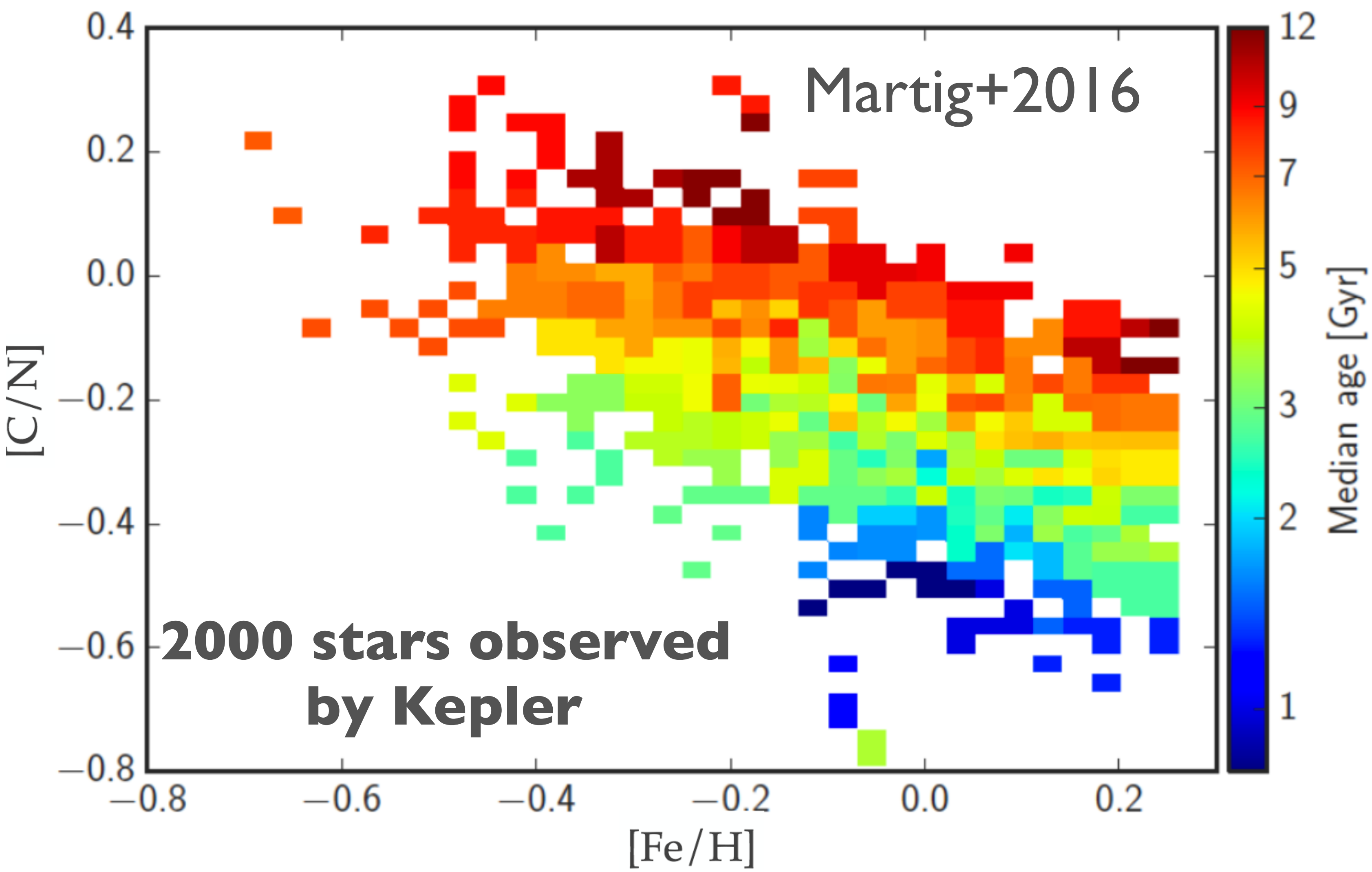
Origin of mass information in the red giants

mass-dependent dredge up changes surface C&N abundance



Origin of mass information in the red giants

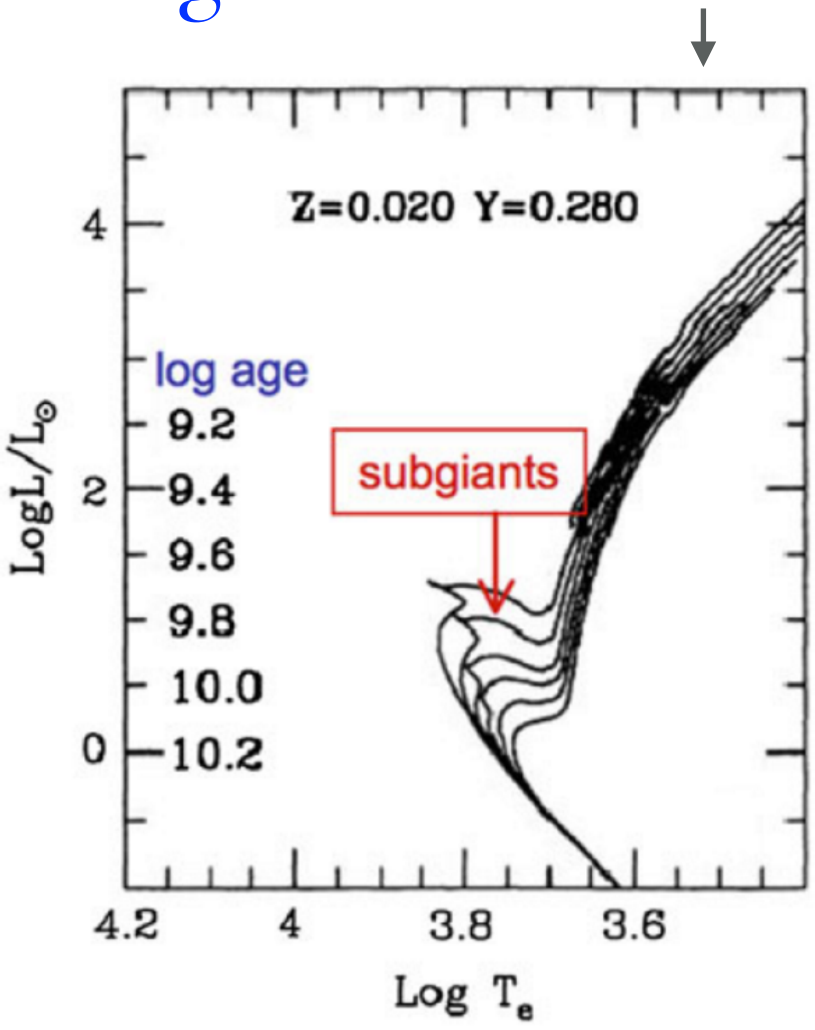
mass-dependent dredge up changes surface C&N abundance



Regime change
small local samples to spectroscopic stellar ages **throughout the disk**

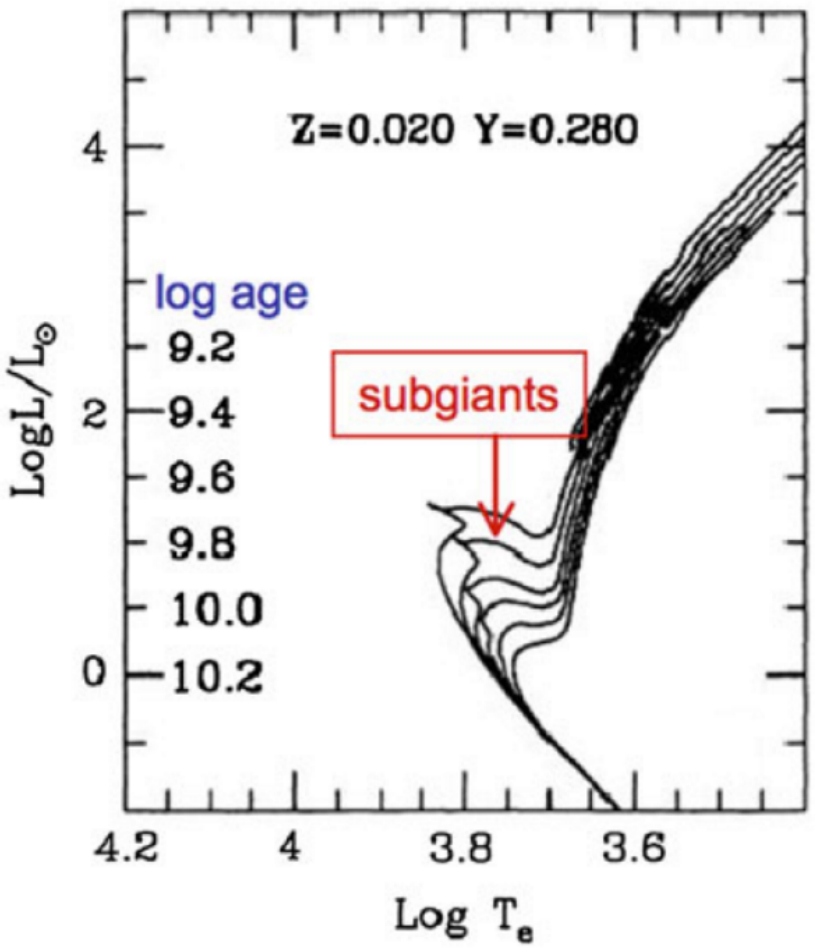
We can now measure ages for giants spectroscopically

regime change:
from stars in the solar neighbourhood....



We can now measure ages for giants spectroscopically

regime change:
from stars in the solar neighbourhood....



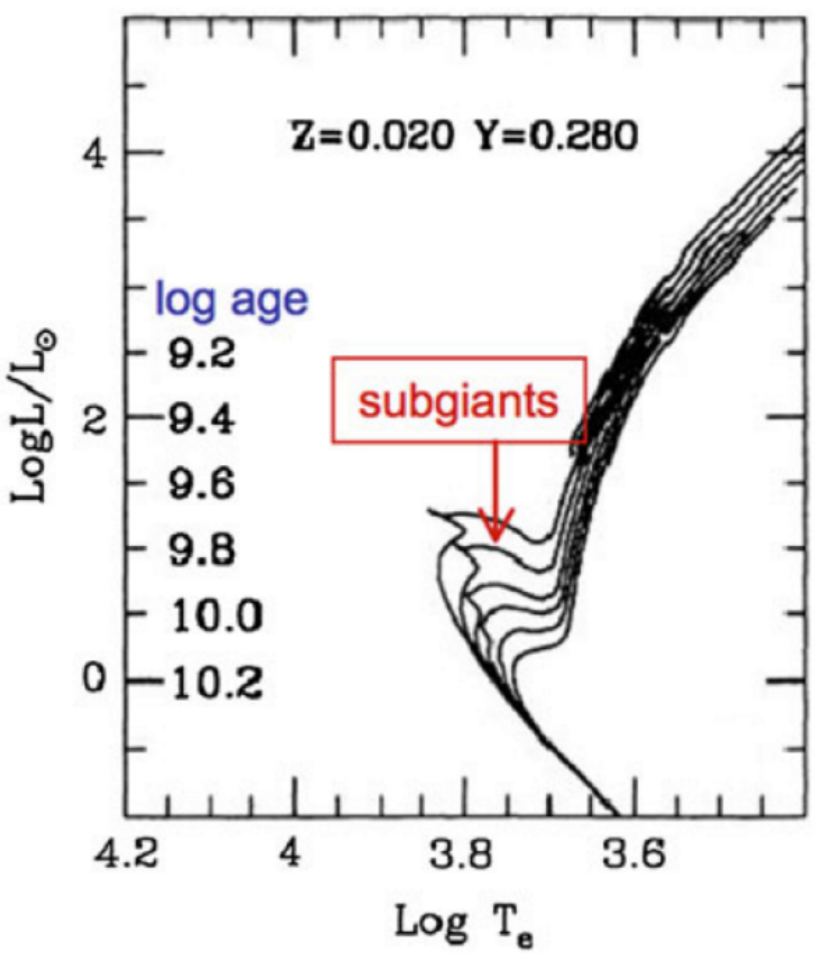
large, vast maps

20kpc

6kpc

We can now measure ages for giants spectroscopically

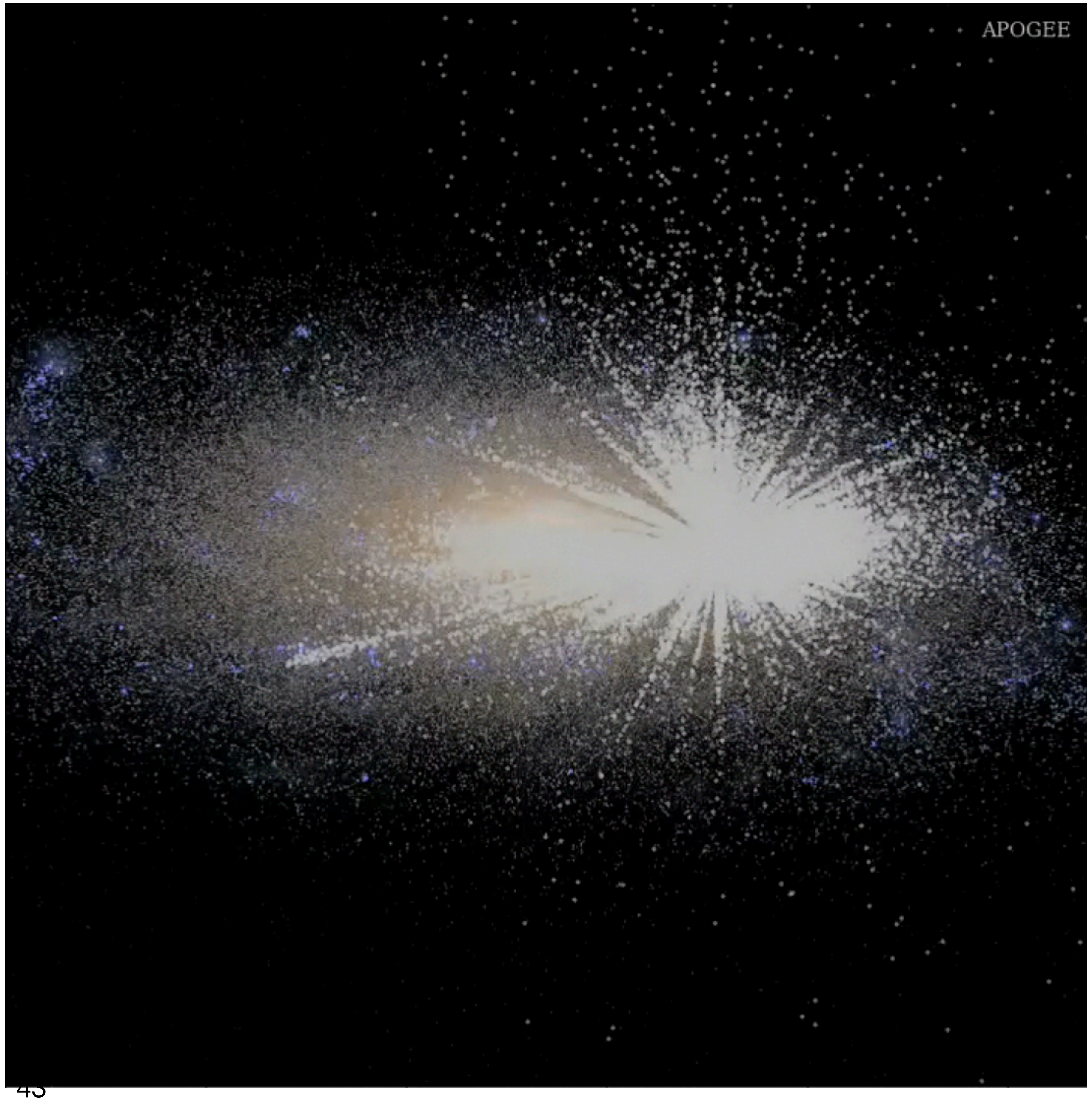
regime change:
from stars in the solar neighbourhood….



large, vast maps

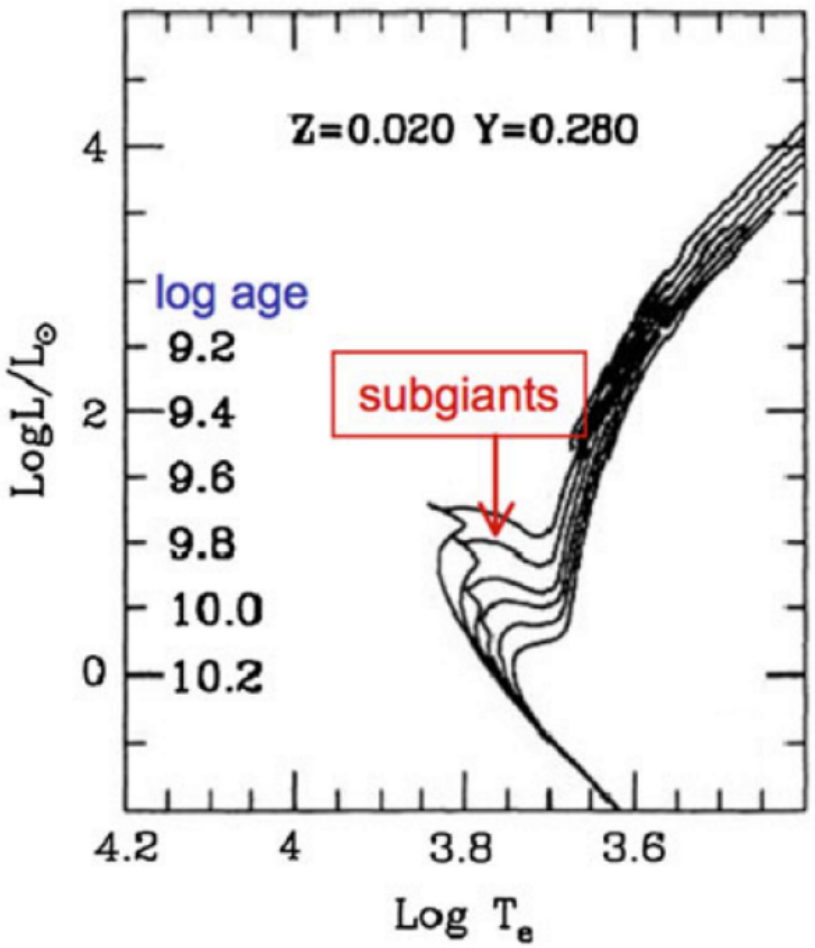
20kpc

6kpc



We can now measure ages for giants spectroscopically

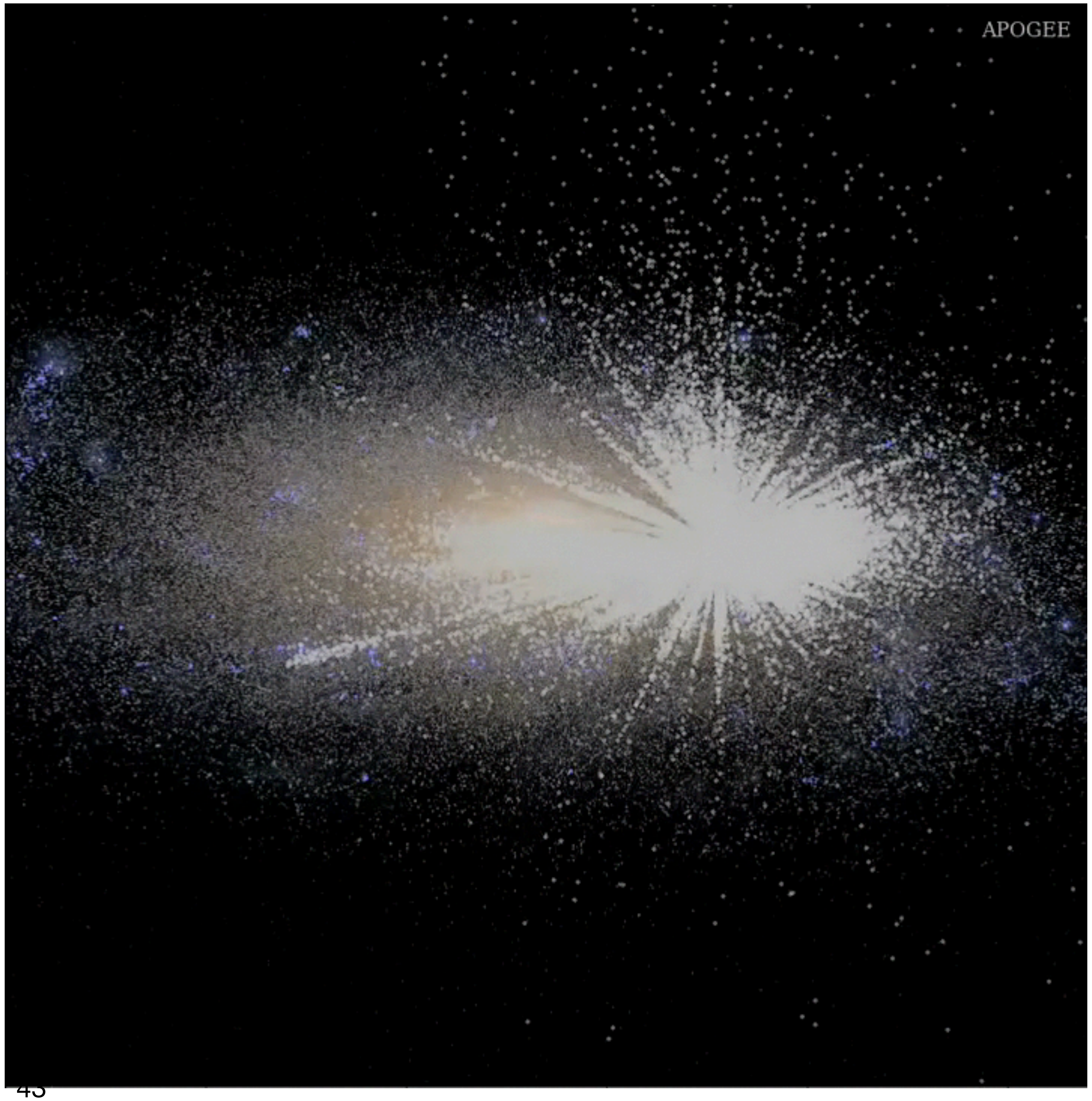
regime change:
from stars in the solar neighbourhood....



large, vast maps

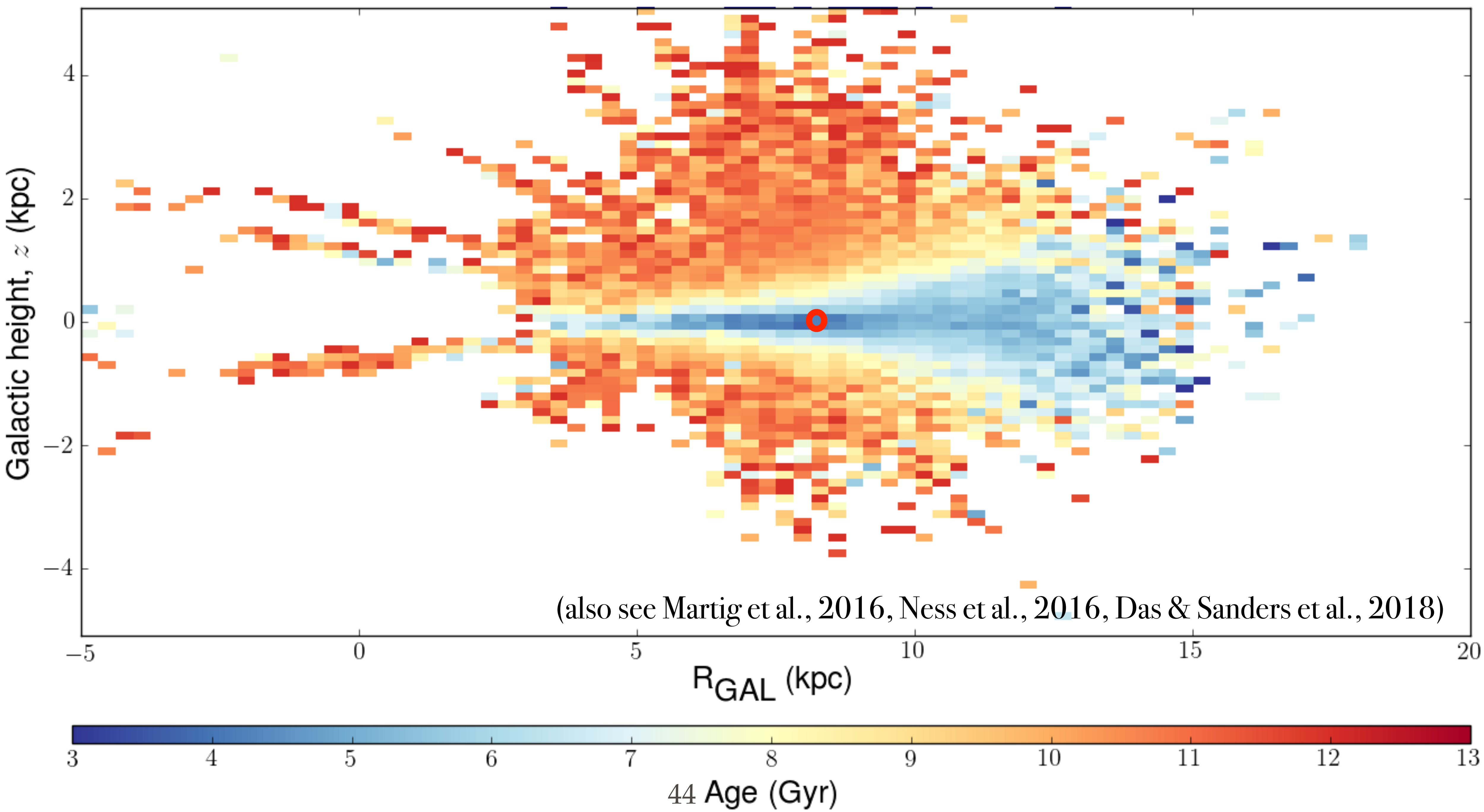
20kpc

6kpc



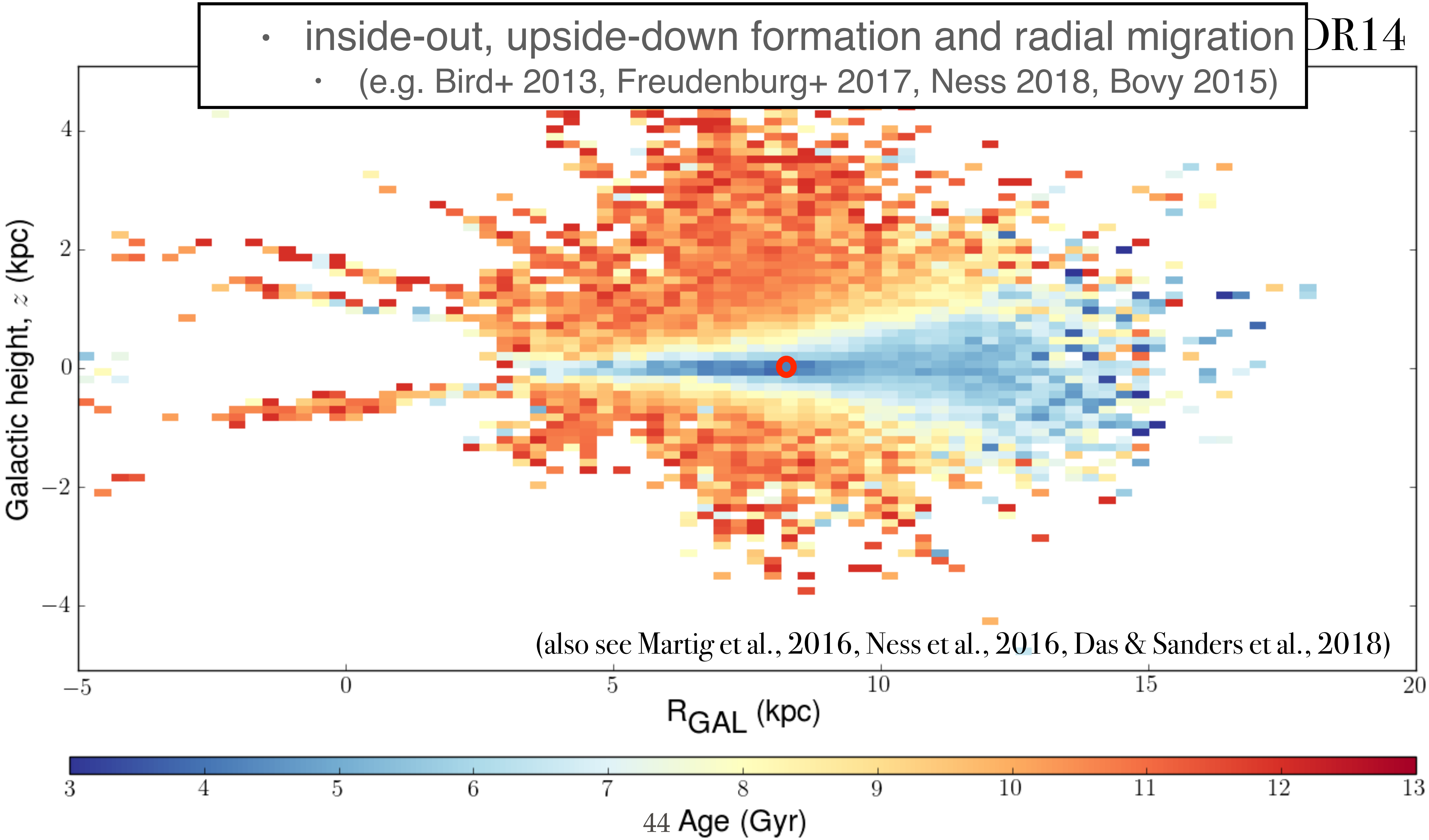
Ages: formation history of disk

150,000 stars from APOGEE DR14

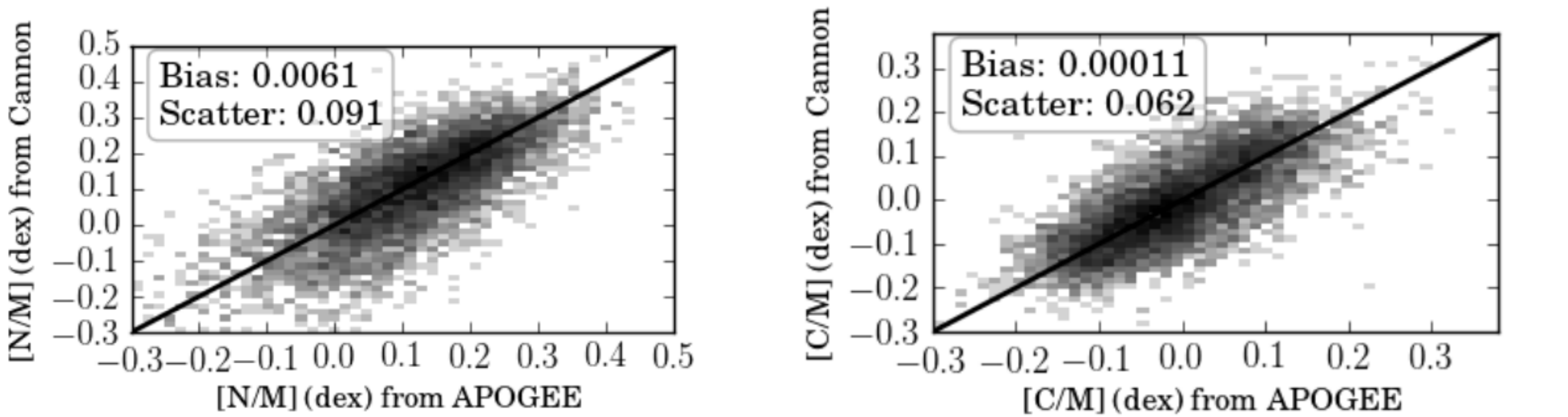


Ages: formation history of disk

- inside-out, upside-down formation and radial migration
  - (e.g. Bird+ 2013, Freudenburg+ 2017, Ness 2018, Bovy 2015)



Learning Carbon & Nitrogen from LAMOST spectra

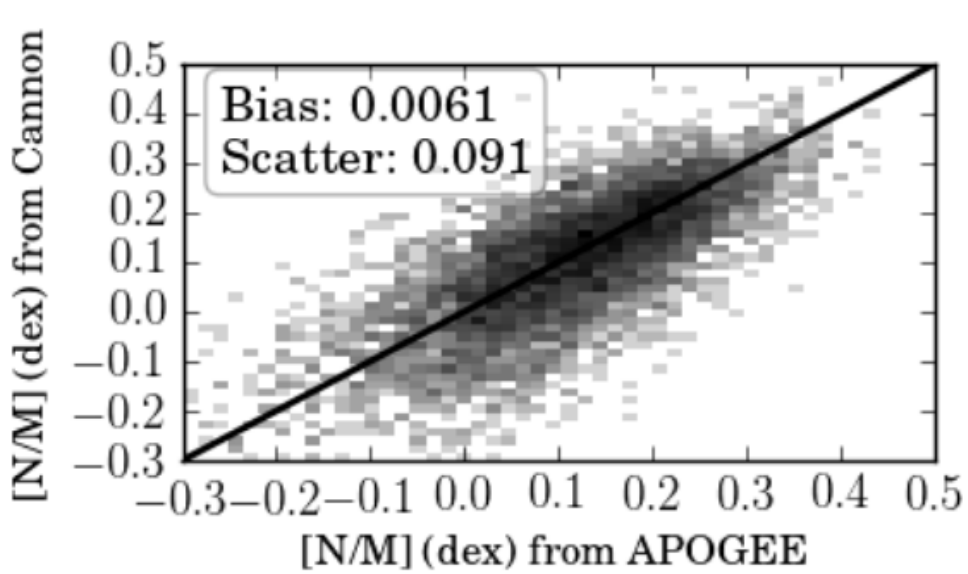


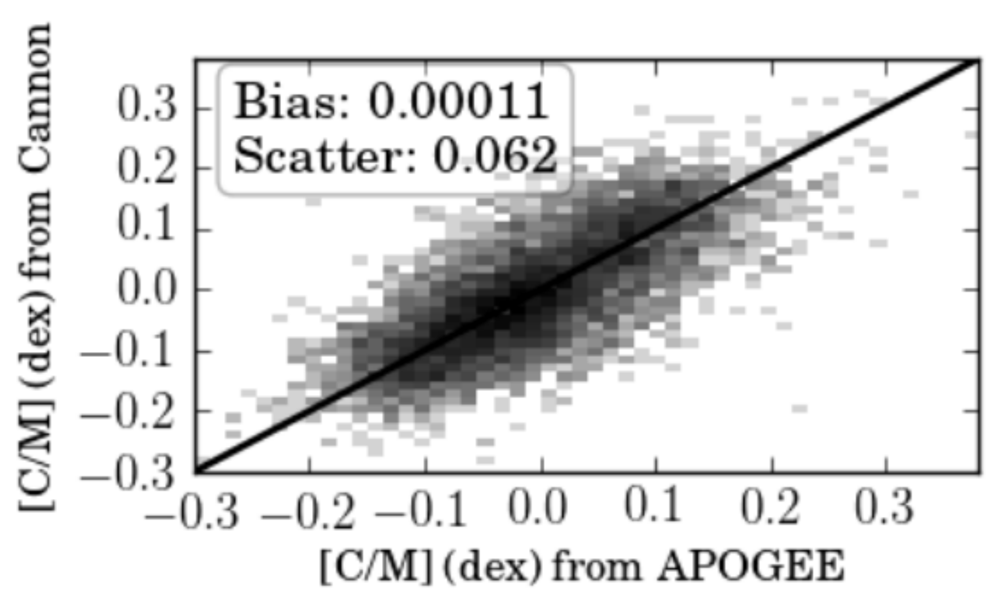
Ho et al., 2016b

C & N ⟶ ages (Martig et al., 2016)

Learning Carbon & Nitrogen from LAMOST spectra

Low resolution spectra is [X/Fe] information rich



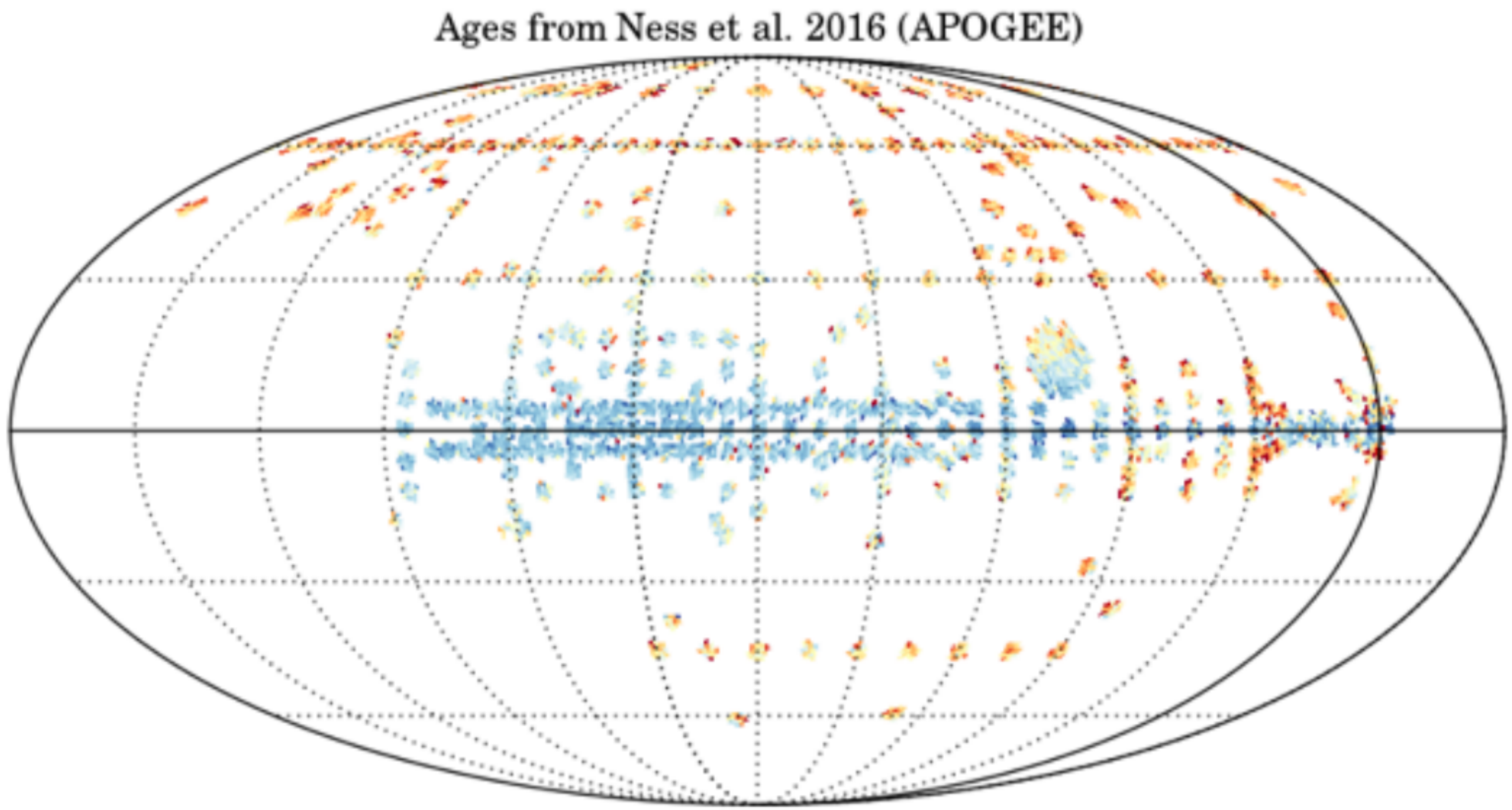


Ho et al., 2016b

C & N $\longrightarrow$ ages (Martig et al., 2016)

APOGEE AGES ON SKY

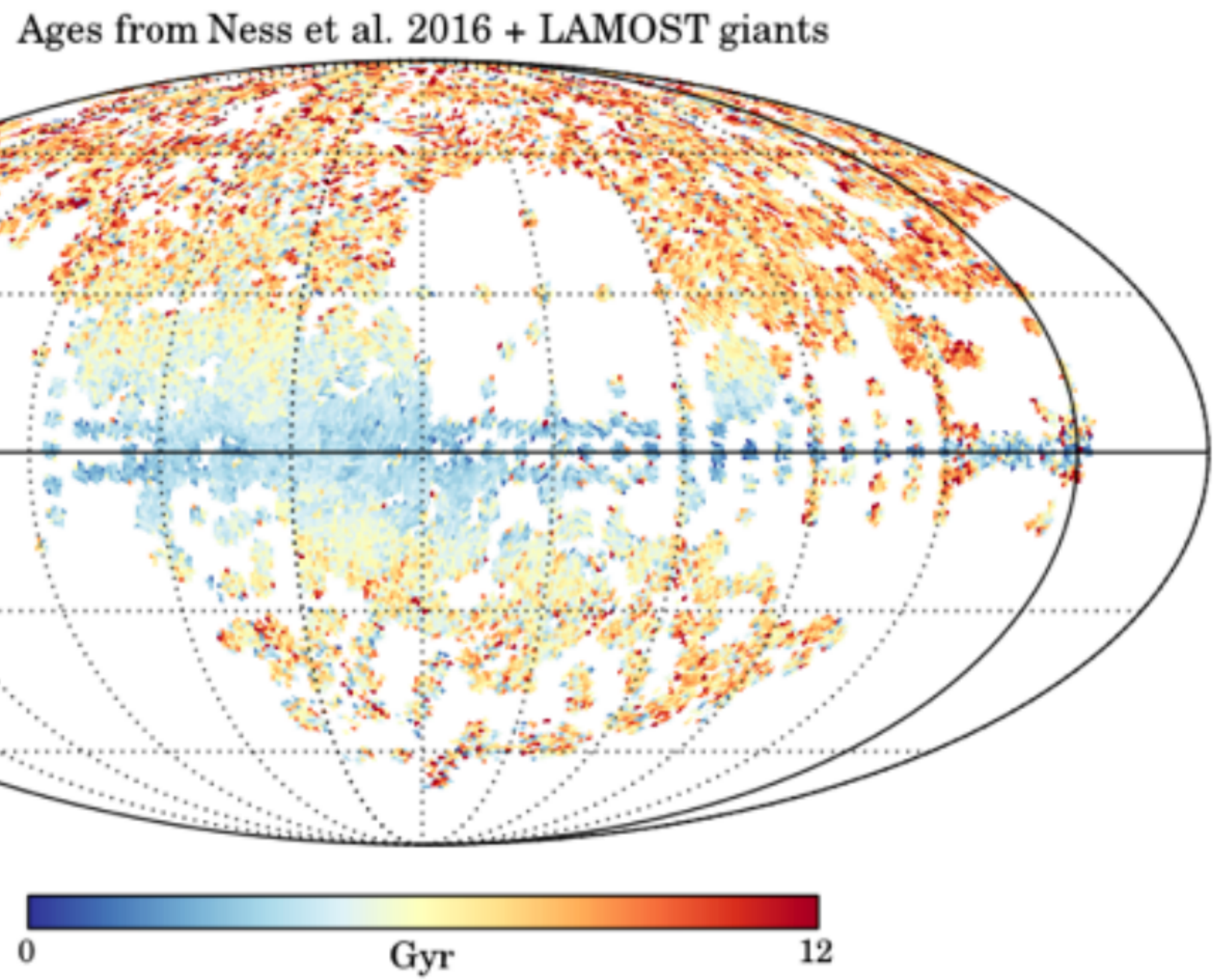
75,000 APOGEE stars



Ho et al., 2016b

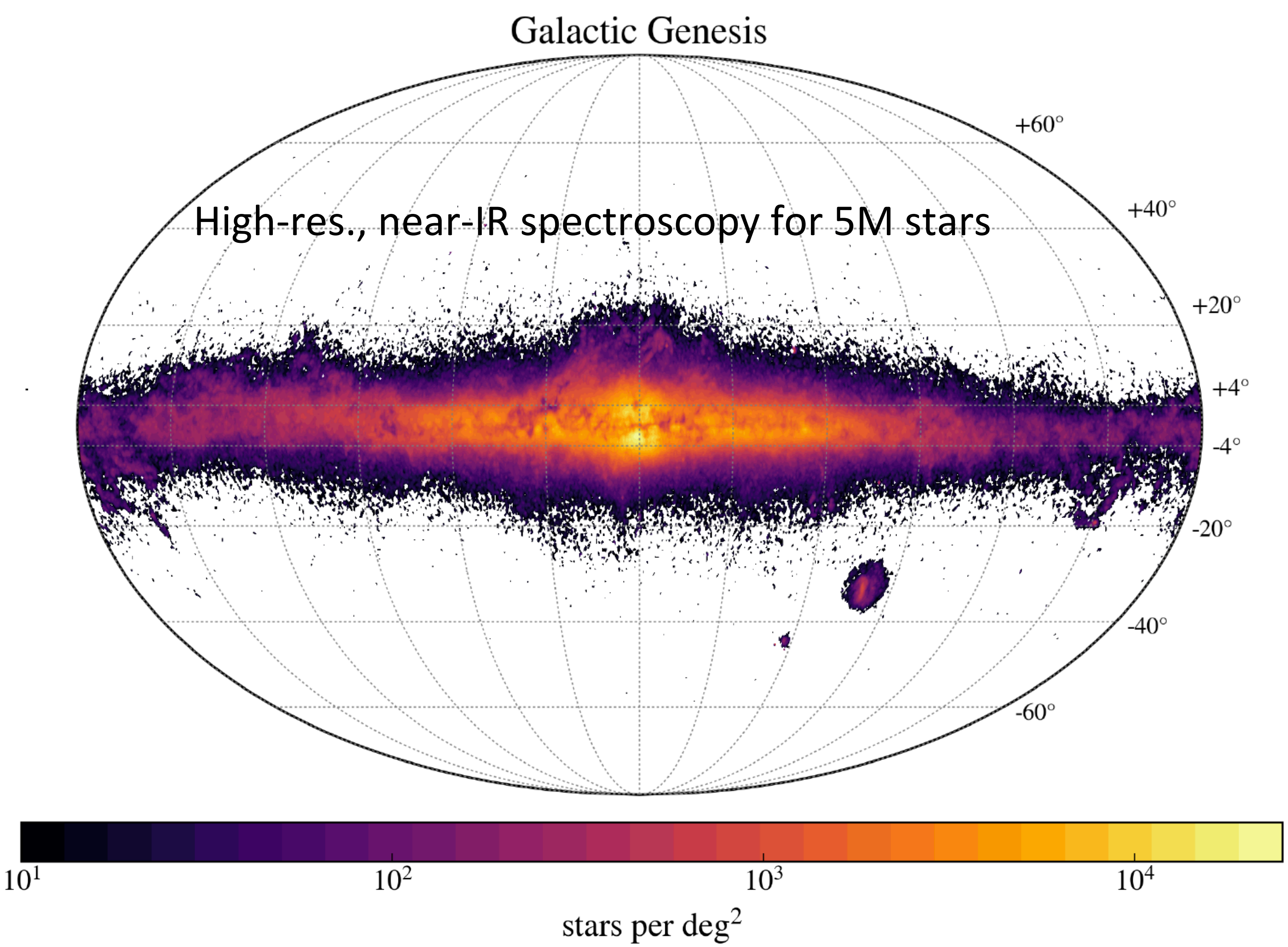
APOGEE + LAMOST AGES

75,000 APOGEE stars
+
230,000 LAMOST stars



Ho et al., 2016b

All the data now is just a warm up for the 2020's



+

Gaia RVS spectra,
higher precision motions