



ILIUM III: Further observations, tests and developments

prepared by: Coryn A.L. Bailer-Jones
Max Planck Institute for Astronomy, Heidelberg
Email: calj@mpia.de

approved by:

reference: GAIA-C8-TN-MPIA-CBJ-046

issue: 1

revision: 0

date: 2009-04-08

status: Issued

Abstract

I present results of further tests of ILIUM for estimating T_{eff} , $\log g$ and $[\text{Fe}/\text{H}]$ for normal stars from BP/RP. I quantify the robustness of the summary statistics to random resampling of the train/test sets, investigate how the performance depends on the wavelength coverage of the input spectra and examine the impact on performance of systematic flux errors in the fluxes. I describe how randomly sampling the grid for building the forward model can bias ILIUM and also explain why it yields systematic errors in the absence of a signature in the data (and I argue why this is actually desirable). I further introduce two modifications to the basic algorithm. First, in an attempt to offset the negative impact of including noisy but AP-sensitive pixels in the input space, a SNR-based weighting scheme is proposed and tested, although it is found to be of limited effectiveness. Second, I propose and demonstrate a method for estimating the covariances (uncertainties) in the estimated APs based on the photometric errors.

Contents

1	Introduction	4
2	Systematic errors and priors with ILIUM	4
3	An erratum and considerations of forward model fitting	6
4	Further tests	10
4.1	3x oversampled data	10
4.2	Robustness of performance statistics under train/test set resampling	10
4.3	Wavelength range selection	12
4.4	Effect on performance of systematic flux errors	17
5	Minor developments	20
5.1	SNR weighting of the sensitivities	20
5.2	AP covariance estimates based on the photometric uncertainties	27
5.2.1	Background material	27
5.2.2	Application to ILIUM	28
5.2.3	Results	30

1 Introduction

I introduced ILIUM in CBJ-042 and CBJ-043 as a new algorithm for estimating parameters from multidimensional data. In those documents I applied it to estimating $T_{\text{eff}}-\log g$ and $T_{\text{eff}}-[\text{Fe}/\text{H}]$ from BP/RP spectra. Here I make some additional tests and modifications of the algorithm and report on the results. The new tests address the following issues

1. whether 3x oversampled spectra improves the AP accuracy over non-oversampled
2. how robust the summary statistics are to random reselection of the train/test sets
3. dependence of AP accuracy on the wavelength range selected
4. effect of systematic flux errors on performance.

I also develop and test two modifications to the algorithm

1. introduction of a SNR weighting term in the AP update equation. This is motivated by the fact that the sensitivities only measure the *ideal* sensitivity of the bands to the APs in the absence of noise
2. an alternative scheme for estimating AP uncertainties. It uses the sensitivity matrix to translate the photometric errors (more generally, covariance) into covariance estimates for the APs.

I also discuss two aspects of the basic algorithm. In section 2 I explain why ILIUM produces systematic errors in AP estimates when there is no signature in the data and how reflects an implicit prior. In section 3 I show how random selection of train and test sets can significantly degrade the forward modelling due to a biasing issue.

2 Systematic errors and priors with ILIUM

Standard machine learning methods are normally trained by minimizing some global error function. With such methods we frequently see systematic errors in those parts of the AP space where the data have no or very low sensitivity to the APs. If there is no signal, the algorithm should ideally report “prior” value for the AP, e.g. the mean in the training set. This is desirable: if the data provide no information, then rather than report an arbitrary value report the best estimate based on what we do know from the training data. This explains why methods such as SVMs and ANNs systematically overestimate the metallicity of low metallicity hot stars (which show no metallicity signature): the mean from the training set is higher than the true value.

In Figure 5 of CBJ-043 we see that ILIUM does the same thing. Why is this, given that it doesn’t minimize a global error? The reason is probably that at low metallicity for hot stars,

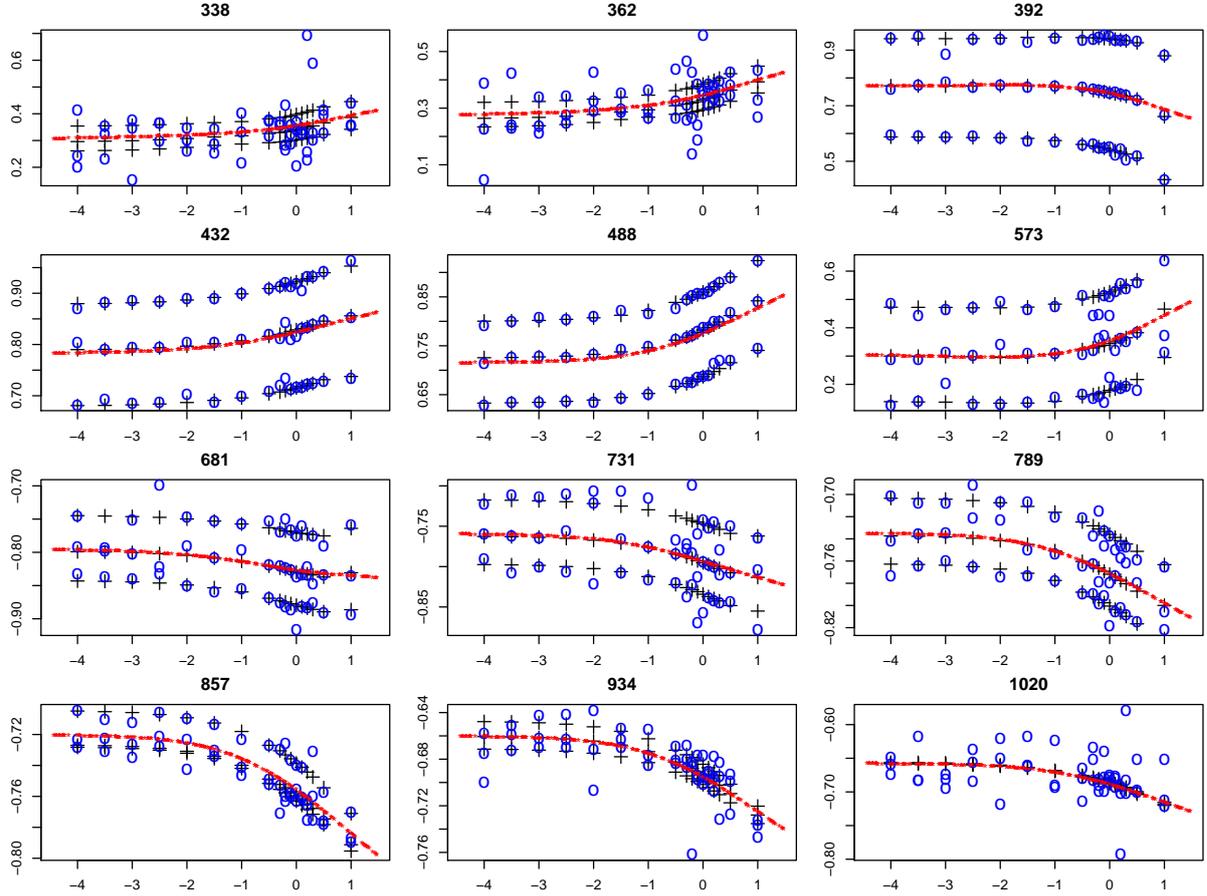


FIGURE 1: Predictions of the full forward model for the dwarfs as a function of $\log(T_{\text{eff}})$ at constant $T_{\text{eff}}=10\,000$ K in 12 different bands (with wavelength in nm at the top of each panel). The black crosses are the (noise-free) grid points, the red stars are the forward model predictions (at randomly selected AP values) and the blue circles the noisy ($G=15$) grid points. The three different blue and black points at each $[\text{Fe}/\text{H}]$ value correspond to the three different $\log g$ values. The flux plotted on the ordinate is in standardized units.

the sensitivity is zero for all or most bands. In this case, the weak forward model provides no increment for the flux at each iterative update. The flux is predicted well by just the strong forward model, which was constructed by averaging over all the metallicities in the training grid. Thus the metallicity predicted by ILIUM will be that average metallicity. This is obviously higher than the lowest metallicity, so we get a systematic error. I can confirm this idea by looking at plots of the forward model. Fig. 1 shows the weak forward model for some bands at $T_{\text{eff}} = 10\,000$ K. The sensitivity (gradient of the forward model) is essentially zero in all bands at $[\text{Fe}/\text{H}]$ below about -2 . This contrasts to the forward model at $T_{\text{eff}} = 5\,000$ K (Figure 4 of CBJ-043): there the sensitivity is significantly non-zero in many bands even at the lowest metallicities. The fact that the systematic error also vanishes in the residuals for just cool stars

(see Figure 6 of CBJ-043) also suggests this interpretation is correct.

The conclusion is that the mean of the weak AP over the training data is a prior on the weak AP, but one which only has an effect when there is essentially no signal in the data. Such a prior is desirable, but it's not ideal that the training data distribution determines it (see Bailer-Jones et al. 2008 for some related discussion).

3 An erratum and considerations of forward model fitting

In CBJ-042 and CBJ-043 I stated that the forward model was fitted using the training set (which was typically a randomly selected subset of 50% of the whole data set). This was not the case (as can be seen in the plots of the forward model). I actually used the whole data set for fitting the forward model. The training set was used to select the nearest neighbour for the initialization of the algorithm (and the model applied to the test set, as stated). This is also done throughout the present TN.

This procedure is legitimate, because the a central aspect of the forward modelling approach is to exploit the full density grid available. We would not want to consciously leave out half the data. With conventional machine learning one may consider this as “cheating”, because we are testing the performance on data which was used to build the model. (It wouldn't be fully cheating, because we train the model on noise-free data whereas we test on noisy.) This would be unacceptable when using template matching algorithms (nearest neighbours), because there we match observations directly with templates, so having the exact template is obviously cheating! But with algorithms which smooth over the training data this is more legitimate, especially if (as we do with ILIUM) we protect against overfitting. The templates are used to learn the overall shape of the function, but they are not used individually in the AP assignments. This is not so different from leave-one-out cross validation with, say, 1000 objects. There we build 1000 models, each trained on 999 objects and the performance assessed on the one left out. Leaving out the one object will hardly affect the model fit, yet we consider this a legitimate procedure.

What's important is how well the algorithm learns the underlying function and how we test this. Distinct train and test sets is a common approach to this, but it may not always be necessary or sufficient. A more significant issue is that the train/test data may be different from the real data, what I call the synthetic spectra mismatch problem (CBJ-044). In CU8 we have also tried training on a random grid and testing on a nominal grid. But this is not fundamentally different from ILIUM, because the random (test) grid is built by interpolating the nominal (training) grid.

ILIUM requires an adequate sampling of the APs to enable it to produce good fits. The required sampling (number and distribution of points) depends on the fitting method used (currently I use smoothing splines with various degrees-of-freedom: see section 3.2 of CBJ-042). What is

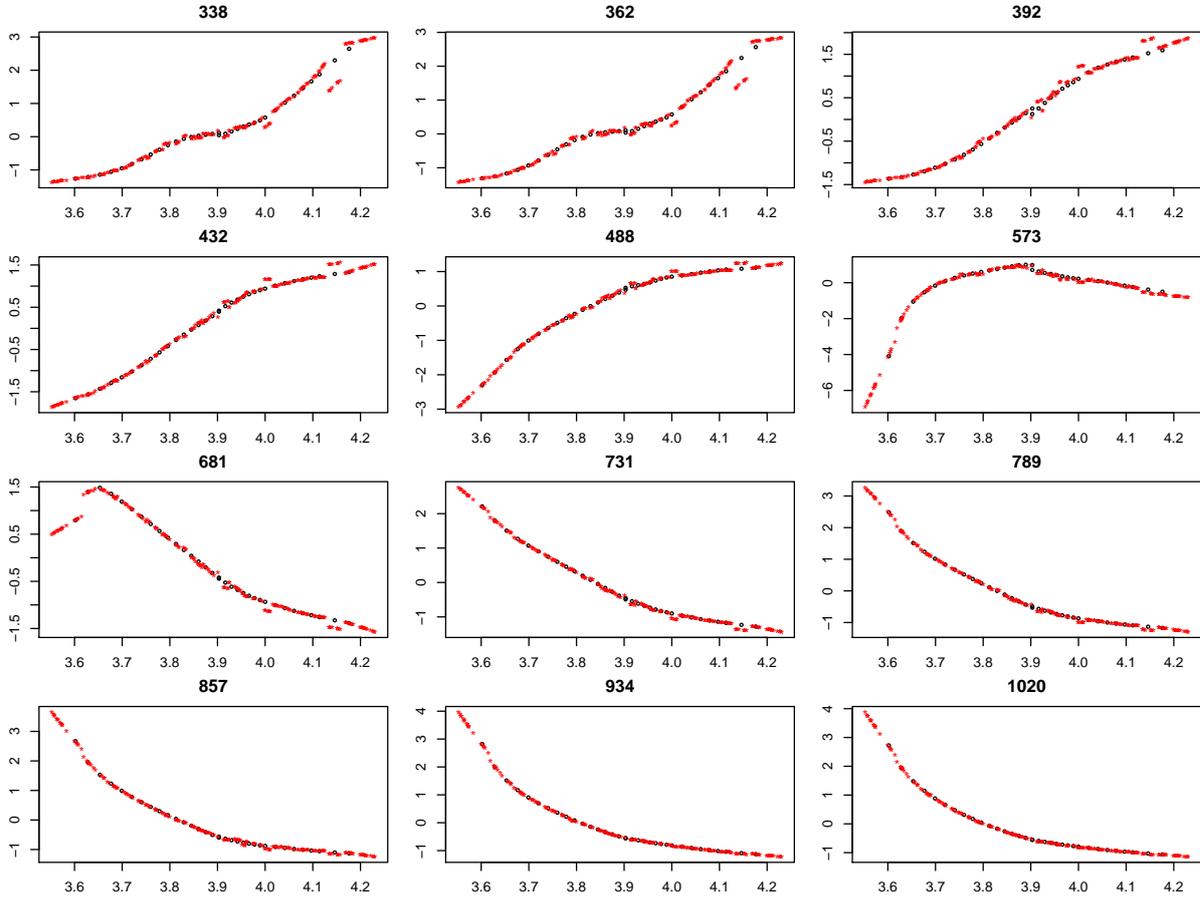


FIGURE 2: Predictions of the *full* forward model (red stars) as a function of $\log(T_{\text{eff}})$ at constant $\log g=4.0$ in 12 different bands (with wavelength in nm at the top of each panel) using just the training data. The black circles are (noise-free) grid points at all $\log g$ used to fit *just the strong component*. The flux plotted on the ordinate is in standardized units.

the impact of using just the training grid (Figure 5 of CBJ-042) for fitting the forward model? Fig. 2 and Fig. 3 show the resulting forward model fits and should be compared to Figures 6 and 7 in CBJ-042 (the corresponding fits using the full data set). We see that the $\log g$ fit is still very good (the shape has hardly changed). In contrast, the T_{eff} fit is broadly similar, but now shows several discontinuities. Why is this?

The forward model is the sum of the strong (T_{eff}) and weak ($\log g$) components. The red points in the plots show the predictions of the full (i.e. summed) forward model with respect to one AP holding the other constant. They are *not* the fits of the individual components. In Fig. 3 the weak component is formed by fitting to the black points. The full forward model is the sum of this plus the fit from the strong component, which is just a constant for this plot (T_{eff} constant). Thus the red curve is offset from what we would expect for a fit to the black point (best seen

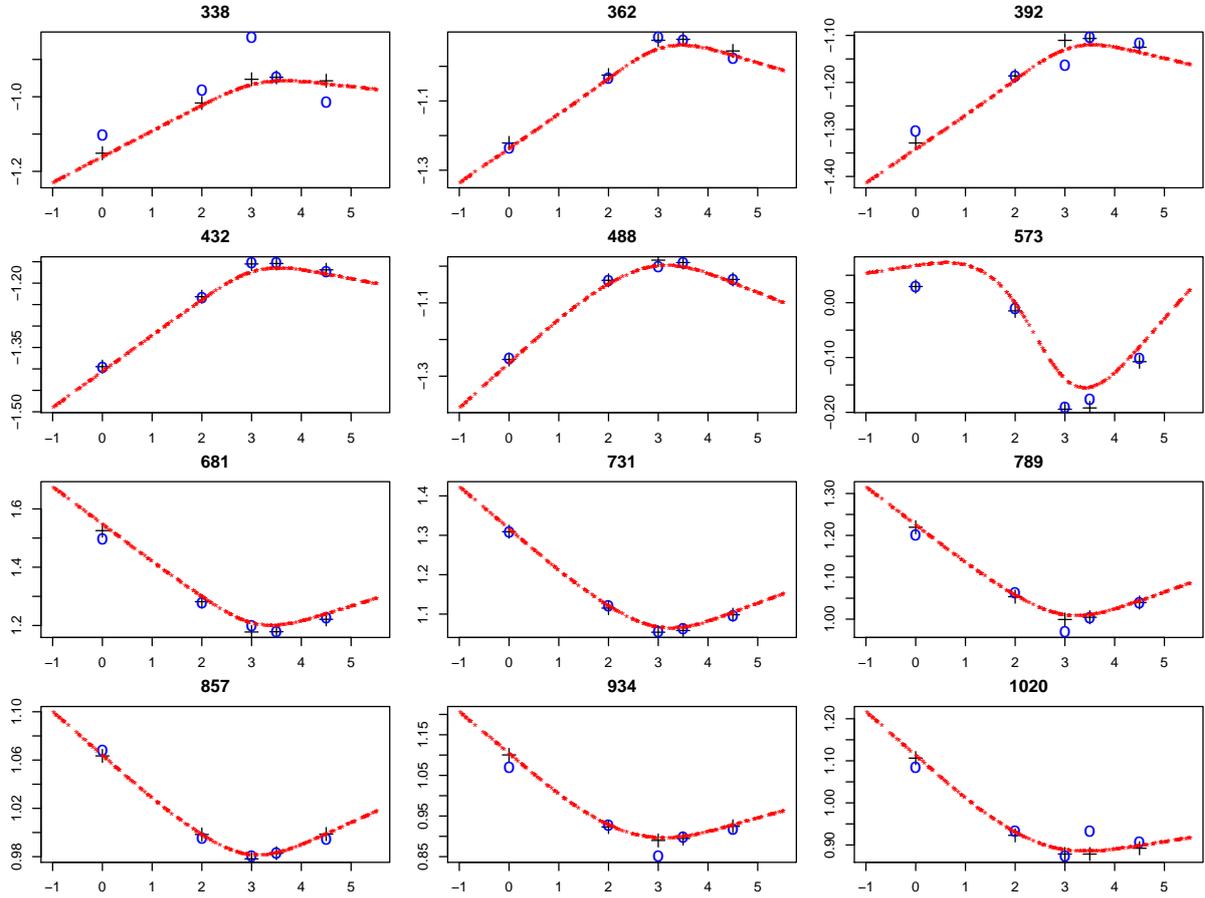


FIGURE 3: As Fig. 2, but now showing predictions of the *full* forward model as a function of $\log g$ at constant $T_{\text{eff}}=5000$ K (red points) and the points at this T_{eff} used *just to fit each weak component* (black crosses).

in the plot for the band at 573 nm). This offset is present because the strong component is a *fit* over the full T_{eff} range: it does not predict the flux at each T_{eff} exactly.

The same affects Fig. 2. Again this plots the full forward model prediction, not the strong component fit. The latter is smooth (it's a spline) whereas the former is this plus an increment at each T_{eff} which depends on the $\log g$ value (here it is $\log g=4.0$). At each T_{eff} value in the training grid we have different $\log g$ values (the blue points in Figure 5 of CBJ-042) so the fit and thus the flux prediction for $\log g=4.0$ does not necessarily change smoothly with T_{eff} . We see strong discontinuities at $T_{\text{eff}} = 10\,000$ K ($\log(T_{\text{eff}})=4.0$) and at $T_{\text{eff}} = 14\,000$ K ($\log(T_{\text{eff}})=4.15$). The reason for this is clear from Figure 5 of CBJ-042: at these two T_{eff} values there are only two low $\log g$ values in the training set, very different from the $\log g$ values used in the weak component fits at the neighbouring points. Moreover, with just two points the forward model fitting algorithm fits a straight line (Fig. 4), so the predictions at $\log g=4.0$ are

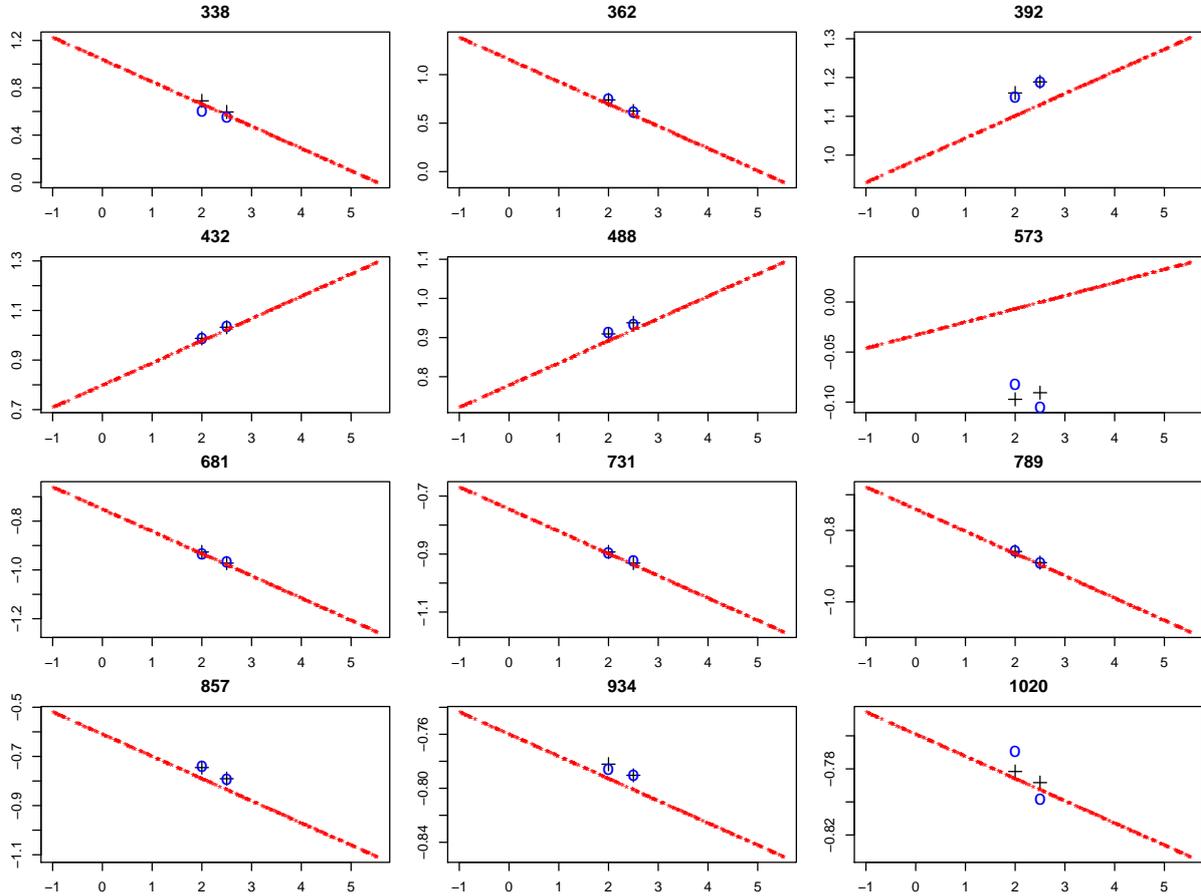


FIGURE 4: As Fig. 3 but now for $T_{\text{eff}}=10000$ K.

poor. The bottom line is that ILIUM was not able to generalise adequately due to a mismatch between training and test sets. The consequence is that when trained on this randomly sampled and sparse training set, ILIUM does very badly on both APs for certain parts of the AP space.

In principle this is also an issue when training on the full grid, but it was much smaller (so hardly degraded the results) because (1) the $\log g$ distribution changed smoothly over the grid (all points in Figure 5 of CBJ-042) and (2) there were enough points to get a good fit. Thus the full forward model varying with respect to temperature is smooth (Figure 6 in CBJ-042) and with respect to gravity has negligible offset (Figure 7 in CBJ-042).

In conclusion: ILIUM not only needs an adequate sampling of both APs to get good 1D fits, but the grid sampling in the weak AP at each value of the strong AP should not be biased. Random sampling of the APs to split into train and test sets can introduce such a bias.

4 Further tests

4.1 3x oversampled data

Some of the experiments in CBJ-043 have been re-run, but now using the 3x oversampled data. (The sigmas are of course kept consistent with this change in the sampling: see section 3.1.2 of CBJ-042.) The summary results are as follows

	[Fe/H]	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	$8.6e^{-3}$	$6.3e^{-5}$	ILIUM, 3x oversampled dwarfs, $G=15$, $T_{\text{eff}} \leq 7000$ K
$ \overline{\delta\phi} $	0.14	0.0017	
σ_ϕ	0.24	0.0023	

For comparison, the results on the non-oversampled data for the same train/test sets from CBJ-043 were

	[Fe/H]	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	$5.6e^{-3}$	$5.2e^{-5}$	ILIUM, dwarfs, $G=15$, $T_{\text{eff}} \leq 7000$ K
$ \overline{\delta\phi} $	0.14	0.0017	
σ_ϕ	0.24	0.0023	

At $G=18.5$ with oversampling we get

	[Fe/H]	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.026	$-1.6e^{-4}$	ILIUM, 3x oversampling, dwarfs, $G=18.5$, $T_{\text{eff}} \leq 7000$ K
$ \overline{\delta\phi} $	0.30	0.0027	
σ_ϕ	0.429	0.0034	

and the results on non-oversampled spectra from CBJ-043 are

	[Fe/H]	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.037	$-4.5e^{-6}$	ILIUM, dwarfs, $G=18.5$, $T_{\text{eff}} \leq 7000$ K
$ \overline{\delta\phi} $	0.26	0.0024	
σ_ϕ	0.42	0.0033	

Oversampling does not improve the averaged performance at either $G=15$ or $G=18.5$ on the $T_{\text{eff}}-[Fe/H]$ problem. This only involved normal stars: emission line stars with resolved features may well benefit from the oversampling (see Lopez Marti et al. 2009 for numerous examples).

4.2 Robustness of performance statistics under train/test set resampling

The tests performed in CBJ-042 and CBJ-043 randomly split the available data into non-intersecting train and test sets. However, the performance on the test set could depend upon the particular train and test sets selected. (As explained in section 3, the training set effects the nearest neighbour initialization and the test set is what we test the performance on, but the forward model is not affected.) To examine this, I reran ILIUM 15 times with randomly resampled train and test sets, using the $T_{\text{eff}}-\log g$ problem at $G=18.5$ at $[Fe/H]=0.0$ (as described in section 3.5 of CBJ-042). There are 137 objects in each of the train and test sets. For each run I calculate

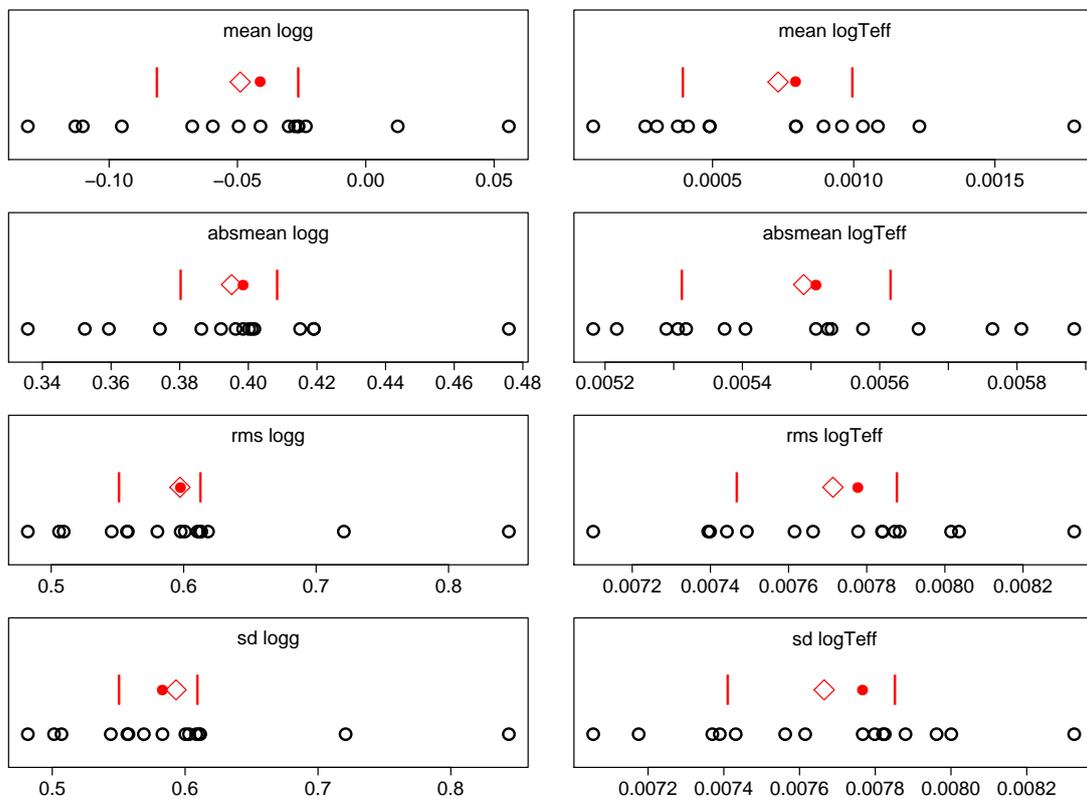


FIGURE 5: ILIUM performance for different randomly selected train and tests on the $T_{\text{eff}}-\log g$ problem at $G=18.5$ at $[\text{Fe}/\text{H}]=0.0$. The individual runs are shown as open circles. The mean, median and upper/lower quartiles are plotted as a diamond, filled circle and vertical bars respectively. The four sets of results correspond to the four different statistics show, with $\log g$ on the left and $\log(T_{\text{eff}})$ on the right.

four statistics: (1) mean residual, $\overline{\delta\phi}$; (2) mean absolute residual, $\overline{|\delta\phi|}$; (3) RMS residual, σ_ϕ ; (4) standard deviation of the residuals. (2) and (3) are summaries of the magnitude of the residual, whereas (1) measures the systematic error and (4) the scatter about this (as it is the RMS of the mean-subtracted residuals). The results are shown in Fig. 5. There is certainly some spread, with 2 of the 15 runs yielding unusually large values for σ_ϕ for $\log g$. The spread between the upper and lower quartiles (contains the central 50% of the results) for $\overline{|\delta\phi|}$ is 0.39–0.42 for $\log g$ (about 8% of the mean value) and 0.00530–0.00555 for $\log(T_{\text{eff}})$ (about 5% of the mean). Thus the reported statistics are robust to within about 5–10%, at least for this problem and this sample size. Larger train and test sets would generally produce a smaller spread. Differences between runs of ILIUM much less than this should not be considered significant.

4.3 Wavelength range selection

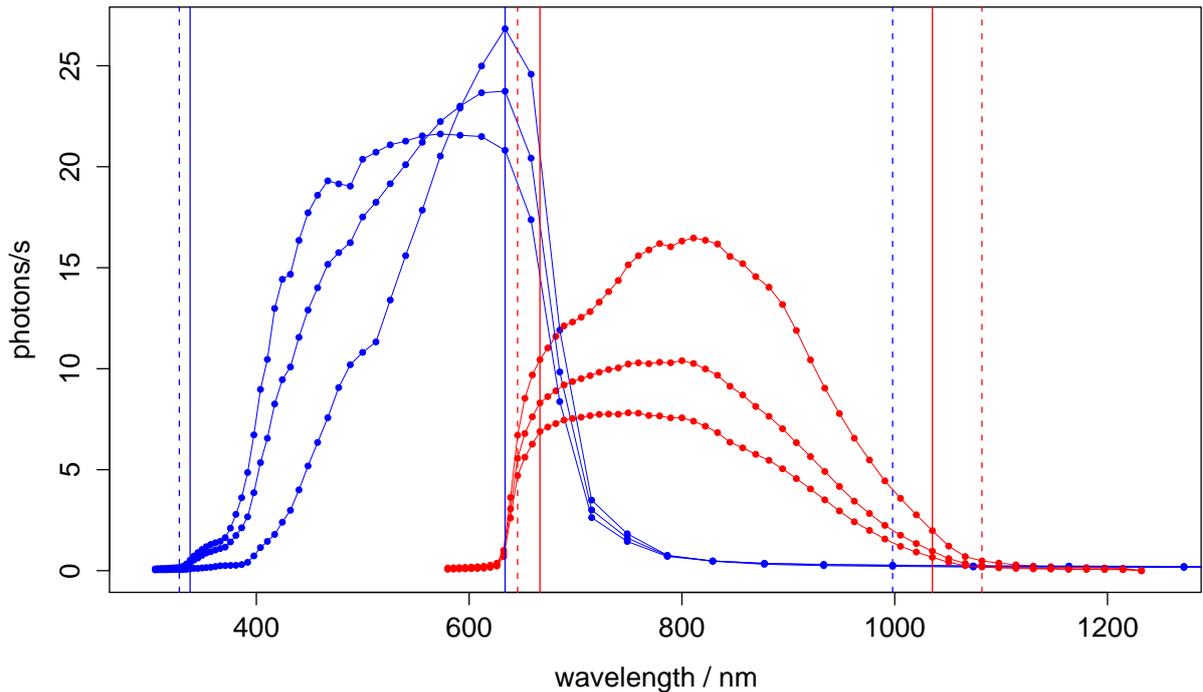


FIGURE 6: BP (blue) and RP (red) noise-free spectra of three stars (A dwarf with $T_{\text{eff}}=10\,000$ K, $\log g=4.0$; F dwarf $T_{\text{eff}}=7000$ K, $\log g=4.0$; K giant $T_{\text{eff}}=4500$ K, $\log g=1.5$; zero metallicity and extinction in all cases). There are 60 pixels in each of BP and RP. These are non-oversampled spectra, and last four pixels of BP (centred at 1272, 1404, 1569 and 1780 nm) are omitted from the plot. The solid vertical lines show the nominal selection range (used in CBJ-042 and CBJ-043). The dashed vertical lines show the wider range used in the DSC-QSO work of Bailer-Jones et al. (2008) and CBJ-036.

The selection of input pixels to use in the model impacts the accuracy of the parametrization,

in the sense that adding more relevant inputs should improve the performance. Each of BP and RP deliver windows with 60 pixels (which I consider as a single 120 element vector) with a wavelength coverage shown in Fig. 6. The nominal wavelength range used here (and in CBJ-042 and CBJ-043) are pixels 12:54, 75:108 (solid vertical lines). The DSC–QSO work of Bailer-Jones et al. (2008) and CBJ-036 used a broader range of 9:54, 72:111 (dashed vertical lines). The narrower range was used for ILIUM because early experiments showed that this improved the results, presumably because this omitted lower SNR pixel bins. Why is this?

Unlike many other machine learning methods, the performance of ILIUM should not suffer due to a higher data dimensionality per se. This is because it fits 1D forward models to each band, so is not subject to the curse of dimensionality in the number of data dimensions (pixels). Because ILIUM uses sensitivity weighting, then including bins which have no or little information (low sensitivity) will receive a low weight in the update equation and have proportionally little influence. However, the sensitivity is calculated using noise-free spectra: the SNR is not taken into account. So if we have bins which are very sensitive to the APs but are typically very noisy, they will get a large weight in the update equation (see section 2.7.4 and Figures 9 and 10 of CBJ-042), but due to noise this update may actually be harmful. Under its present formulation ILIUM could have problems if it uses bins which have a high sensitivity but a low SNR. This could apply to the very blue pixels in BP.

Here I apply ILIUM to the same problem as in section 4.2, but now investigate the effect of varying the pixel range used. I again perform 15 runs with the same 15 random train/test sets used previously. The results are shown in Fig. 7 are for the pixel following ranges and corresponding wavelength ranges (from top to bottom):

1. BPnom: BP only from nominal, 12:45 (338–634 nm)
2. RPnom: RP only from nominal, 75:108 (667–1035 nm)
3. nom: nominal wavelength range, 12:45, 75:108
4. nom+: nominal, plus three more pixels at red end of BP and blue end of RP, 12:48, 72:108 (338–715, 646–1035 nm)
5. nom++: nom+, plus three more pixels at blue end of BP, 9:48, 72:108 (328–715, 646–1035 nm)
6. DSC/QSO paper range, 9:54, 72:111 (328–998, 646–1082 nm)

The nominal range is more or less the best on both APs despite the fact that BP is heavily cut at the the red end (quite near the peak of the sensitivity) and the red is cut in the blue end well before the drop. Adding three more pixels in each case to extend to regions of lower sensitivity (the nom+ result) increases the performance by a very small amount which is barely significant

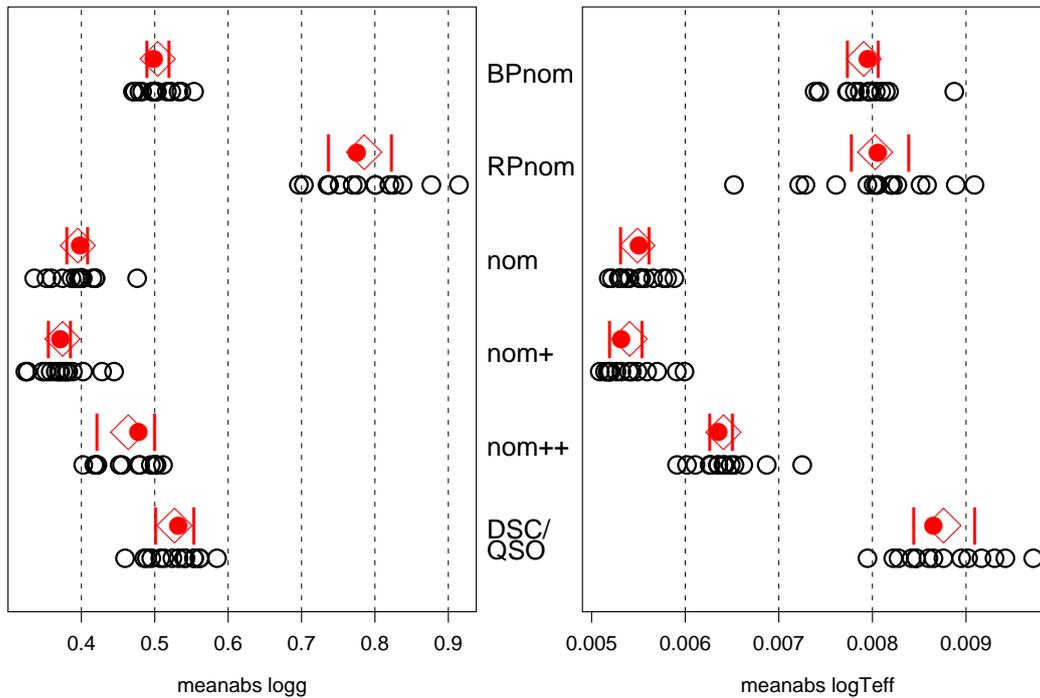


FIGURE 7: ILIUM performance on $T_{\text{eff}}-\log g$ problem at $G=18.5$ at $[\text{Fe}/\text{H}]=0.0$ using different input pixels (described in the text). The performance metric is the mean absolute error over the whole test set. Open circles are the result of different randomly selected train/test sets. The mean, median and upper/lower quartiles are plotted as a diamond, filled circle and vertical bars respectively.

(the inter-quartile ranges overlap). Surprisingly, adding just another three pixels to the blue end of BP (the nom++ result) actually degrades the performance by a statistically significant amount compared to nom for both $\log g$ (by almost 0.1 dex) and $\log(T_{\text{eff}})$ (by 0.001 dex), some 15% of the average performance. This is presumably a result of these pixels having a high sensitivity yet low SNR, thereby adding a highly weighted but almost random update, as described above. This dependence of performance on the wavelength range selected is clearly undesirable. A modification of the sensitivities to accommodate the SNR is described in section 5.1.

Using the broader ‘‘DSC/QSO’’ range degrades $\log(T_{\text{eff}})$ further over nom++, but with a lesser effect on $\log g$. This may again be on account of misleading updates from low SNR pixels. Not surprisingly using just the BP or RP portion of the nominal range (BPnom, RPnom) is worse than using the two together: RP alone is particularly bad at $\log g$ (a factor of two worse). This must be due to the reduction in information.

Note that the nominal wavelength range (used in CBJ-042, -043) omits $H\alpha$ in both BP and RP. The range was selected based on the performance for normal stars, and not emission line stars. Although $H\alpha$ is not visible in BP/RP in absorption (not even in 4x oversampled spectra) it can be seen in emission for Be stars (see Lopez Marti et al. 2009), so once we expand the object-type space, we’ll need to include this.

I now repeat the above, but on the $T_{\text{eff}}\text{--}[\text{Fe}/\text{H}]$ problem for dwarfs at $G=15$ presented in CBJ-043. There I trained and tested ILIUM using stars with T_{eff} spanning the range 4000–15 000 K. $[\text{Fe}/\text{H}]$ was poorly determined for hot stars (a well known phenomenon), so the relevant results were those reported for $T_{\text{eff}} \leq 7000$ K. Including the hot stars in the training set shouldn’t adversely effect the model performance on $[\text{Fe}/\text{H}]$ for cool stars, because the weak components in the forward model are all independent 1D functions of $[\text{Fe}/\text{H}]$ at each grid value of T_{eff} . So a weak or non-existent signature at high T_{eff} does not affect the $[\text{Fe}/\text{H}]$ estimation at low T_{eff} . As we know a priori that we only want to estimate $[\text{Fe}/\text{H}]$ for cool stars it makes little sense to train on hot stars at all. So here I train and test test ILIUM just for stars lying in the range 4000–7000 K (inclusive), with the full $[\text{Fe}/\text{H}]$ range used in CBJ-043 (+1 to -4 dex). These are zero extinction dwarfs with $\log g \in \{4.0, 4.5, 5.0\}$ (same as in section 2.1 of CBJ-043). There are 657 stars in total, which again are randomly split into equal-sized train and test sets.

The results are shown in Fig. 8. Considering first $[\text{Fe}/\text{H}]$, we see that again the nom and nom+ results are the best. The results with the broader nom++ and DSC/QSO ranges are, as before, worse by a *statistically* significant amount but by a *scientifically* insignificant one. Using just BP or RP alone degrades the results by 0.07 dex on average. Contrary to what one might expect, RP does indeed carry significant information on the metallicity. Turning now to the results for T_{eff} , we find some surprising results. Most noticeable is the fact that RP alone does significantly better than nom and nom+, reducing the mean absolute residual by 25%. The performance with BP alone, in contrast, is significantly worse. Including the three very low SNR pixels in BP (nom++ result) now does not degrade the performance for T_{eff} at all, and the even wider range of DSC/QSO does not lower performance over nom as much as it did for the $T_{\text{eff}}\text{--}\log g$ problem

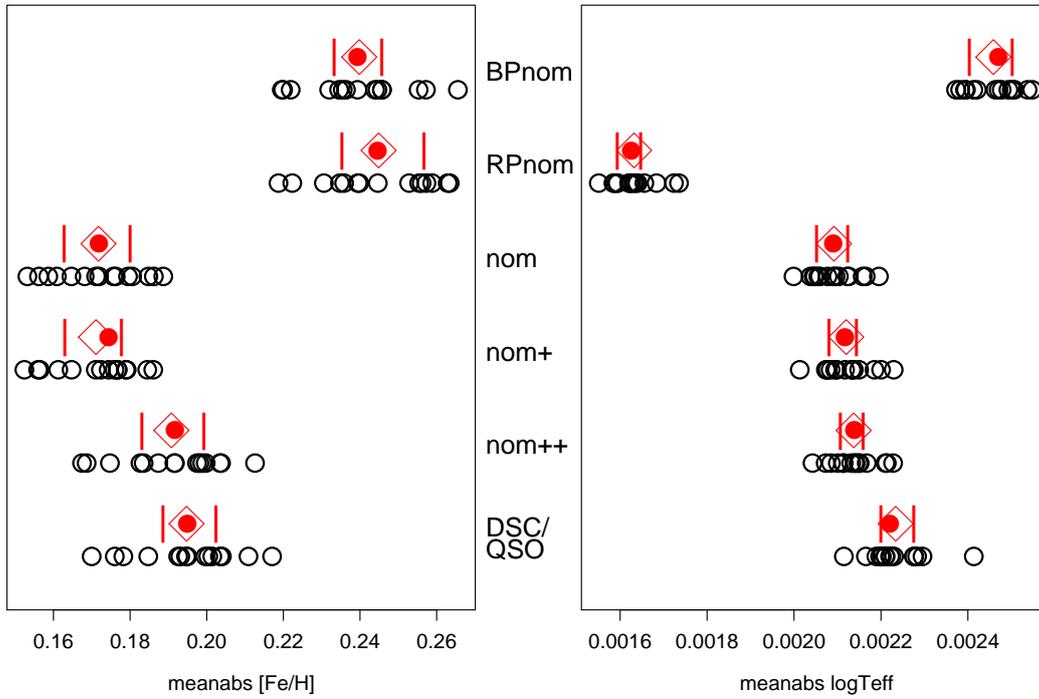


FIGURE 8: ILIUM performance on $T_{\text{eff}}-[Fe/H]$ (for $T_{\text{eff}} \leq 7000$ K) problem at $G=15$ for dwarfs using different input pixels (described in the text). The performance metric is the mean absolute error over the whole test set. Open circles are the result of different randomly selected train/test sets. The mean, median and upper/lower quartiles are plotted as a diamond, filled circle and vertical bars respectively.

(Fig. 7). This might be because we now have only cool stars, and on average the extra pixels in these wider ranges have not very high sensitivities for cool stars. When comparing these results to Fig. 7, recall that we have changed both the T_{eff} range and the G magnitude. Both could explain why the RPnom range alone give the best results: averaged over the sample, the cool star sample has more flux in RP than in BP, so combined with the overall lower SNR, BP may be providing a net confusing sample. This again suggests that we could benefit if we use some form of SNR weighting in ILIUM (see section 5.1).

4.4 Effect on performance of systematic flux errors

The BP/RP data processing will not be perfect. Here I investigate the impact of two errors on the ILIUM with the $T_{\text{eff}}-\log g$ problem at $G=15$ (see section 3.3 of CBJ-042 for the nominal results).

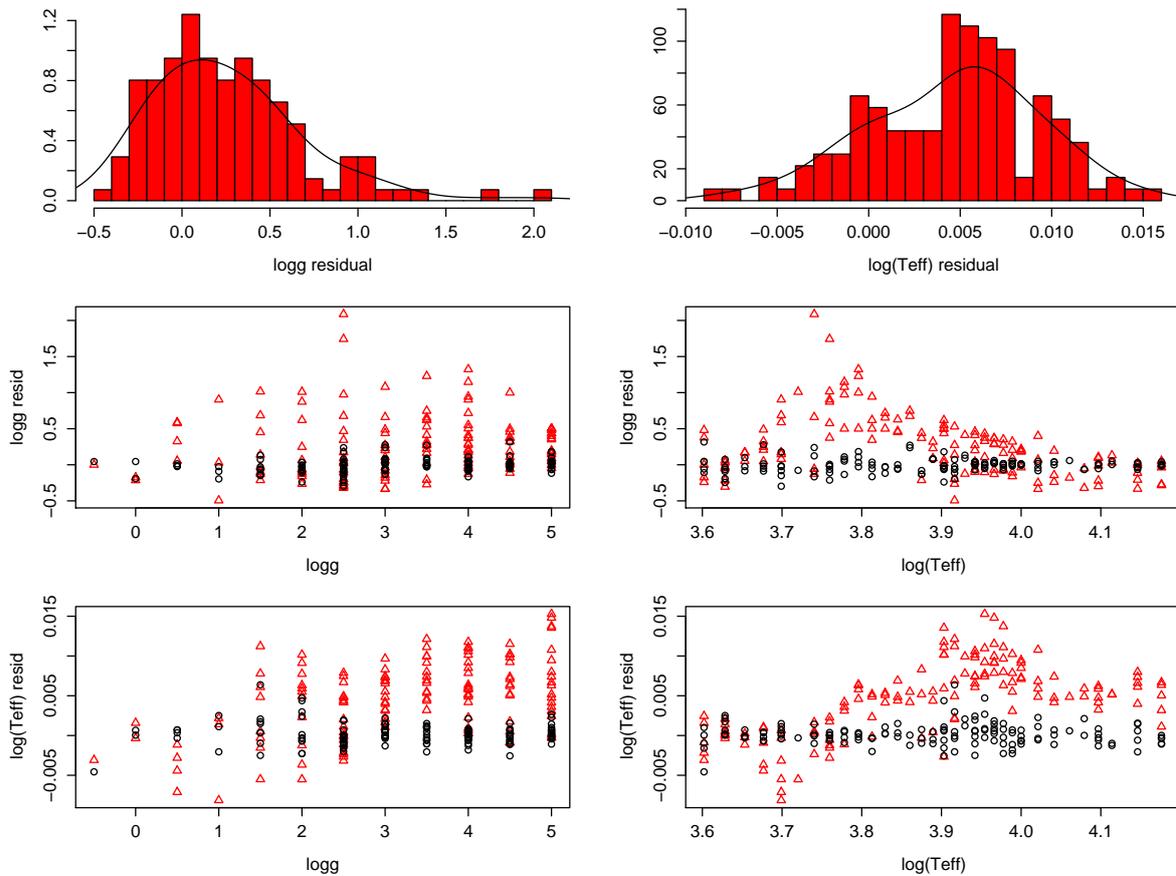


FIGURE 9: AP residuals on the $T_{\text{eff}}-\log g$ problem at $G=15$ with a 1% systematic additive flux error (histograms and red triangles) compared to the unperturbed data (black circles)

Additive error. For each spectrum in the test set I calculate the median flux and add e times this to each pixel. (This is done before standardizing the data.) The training data and forward models are unchanged. This simulates a constant (across the spectrum) background term, thus simulating imperfect background subtraction in the data (whereas the forward model assumes perfect correction). With $e = 0.01$, the residuals are show in Fig. 9 plotted in red, with the unperturbed results (Figure 11 of CBJ-032) shown in black. The performance statistics are

	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	0.026	0.0046	ILIUM, G=15, 1% systematic additive error in fluxes
$ \overline{\delta\phi} $	0.36	0.0055	
σ_ϕ	0.50	0.0065	

Already with a 1% flux error the performance is seriously degraded, by a factor of five in $|\overline{\delta\phi}|$ for both APs. With a 5% error, $|\overline{\delta\phi}|$ in $\log g$ is 1.4 dex and there is a systematic error of 0.6 dex: $\log g$ cannot be estimated! $\log (T_{\text{eff}})$ holds up better with a $|\overline{\delta\phi}|$ of 0.028 dex, but the systematic is almost as large.

Multiplicative error. For each spectrum in the test set I multiply the flux by $(1 + e)$. (This is done before standardizing the data.) The training data and forward models are unchanged. This simulates a multiplicative flux calibration error in each pixel (whereas the forward model assumes perfect calibration). With $e = 0.01$, the residuals are show in Fig. 10 plotted in red, with the unperturbed results (Figure 11 of CBJ-032) shown in black. The performance statistics are

	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	0.057	0.0006	ILIUM, G=15, 1% systematic multiplicative error in fluxes
$ \overline{\delta\phi} $	0.23	0.0019	
σ_ϕ	0.33	0.0024	

There is again a degradation, but it is not as serious as with the additive error. This is because the systematic error added in the additive case will be much larger than in the multiplicative case for pixels with low counts (at the ends of the spectrum), because for them 1% of the median (over the spectrum) is much larger than 1% of their own flux.

That even small systematic errors degrade the performance by so much is not surprising when we consider how ILIUM works. At each iteration it uses the difference in flux between the forward model prediction and the measurement (step 3 of the algorithm; section 2.2 of CBJ-042). If these are on systematically different scales, ILIUM will give poor results. We could mitigate the effect of an additive error by setting the median or mean flux (across the spectrum) to zero for both the observed spectra and the training data. But such fixes should wait until we have a better idea of the magnitude and nature of the calibration errors in the BP/RP data.

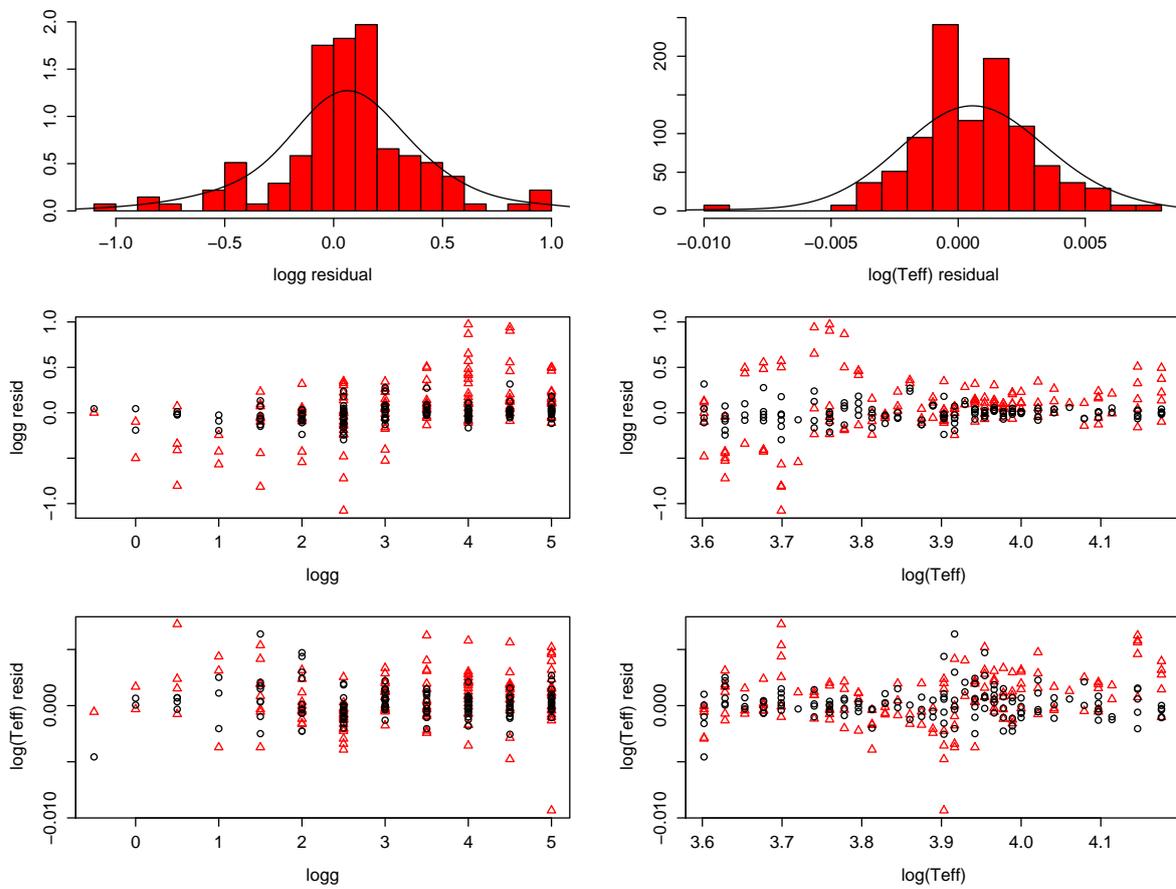


FIGURE 10: AP residuals on the $T_{\text{eff}}-\log g$ problem at $G=15$ with a 1% systematic multiplicative flux error (histograms and red triangles) compared to the unperturbed data (black circles)

5 Minor developments

5.1 SNR weighting of the sensitivities

We saw in section 4.3 that the performance depends on the pixel range used, typically being worse if a broader range was used which includes low sensitivity pixels. I speculated the cause to be quasi-random contributions to the AP updates from high sensitivity but noisy pixels. While sensible feature selection is a prerequisite for machine learning, it is not obvious what the trade-off should be between sensitivity and SNR, or below what SNR level one should drop data. Moreover, it is desirable to use a common set of pixels for classification for most objects, so some active weighting of pixels to account for the SNR – which can usually be estimated – could be beneficial.

An appropriate scheme is to replace the sensitivity, s_{ij} , with $w_i s_{ij}$ where w_i is a weighting factor for pixel i . An appropriate weight is the SNR in pixel i , SNR_i . In order that the weighting doesn't change the overall scale of the updates, I define the weights as

$$w_i = \frac{\text{SNR}_i}{\langle \text{SNR}_i \rangle} \quad (1)$$

Thus equation 3 in CBJ-042 becomes

$$\delta \mathbf{p} = W S \delta \phi \quad (2)$$

where

$$W = \text{diag}(w_i) \quad (3)$$

is a diagonal $I \times I$ matrix. We replace S with WS in the update equation 5 in CBJ-042. The forward modelling is not affected.

I used this SNR weighting method on the nominal $T_{\text{eff}} - \log g$ problem (section 3.3 of CBJ-042). I select five particular stars for analysis. Their AP updates are shown in Fig. 11 and their SNR weighting vectors (spectra) in Fig. 12. The spectra of AP updates for these five stars at the second iteration are shown in Fig. 13. For comparison, Fig. 14 shows the corresponding AP update spectra when not using SNR weighting. We see how the SNR weighting suppresses contributions from the very blue end of the spectrum and, correspondingly, how it enhances contributions in the centre of the spectrum, especially at band 30.

I now carry out a more systematic comparison with and without SNR weighting for different wavelength ranges. Taking into account the variance in the summary statistics due to randomized train/test samples, I perform tests on the same 15 sets as used in sections 4.2 and 4.3. The goal is to assess whether SNR weighting helps when including low SNR pixels, so I do tests on the “nom”, “nom++” and “DSC/QSO” ranges defined in section 4.3.

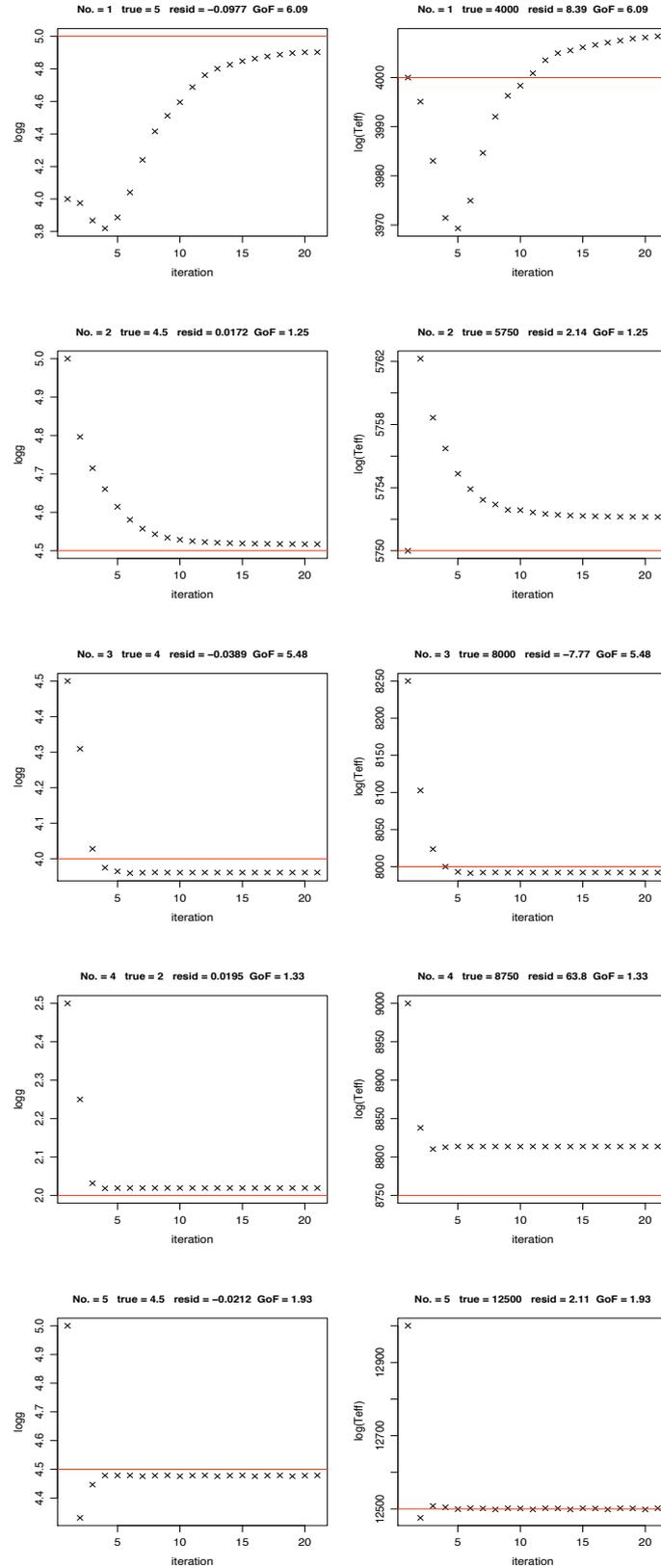


FIGURE 11: AP evolution for five stars in the test data set at $G=15$ ($\log g$ left, $\log(T_{\text{eff}})$ right) using SNR weighting. The true APs are written at the top of each panel pair and plotted as the red horizontal line. Cf. Figure 8 of CBJ-042 which shows the same without SNR weighting.

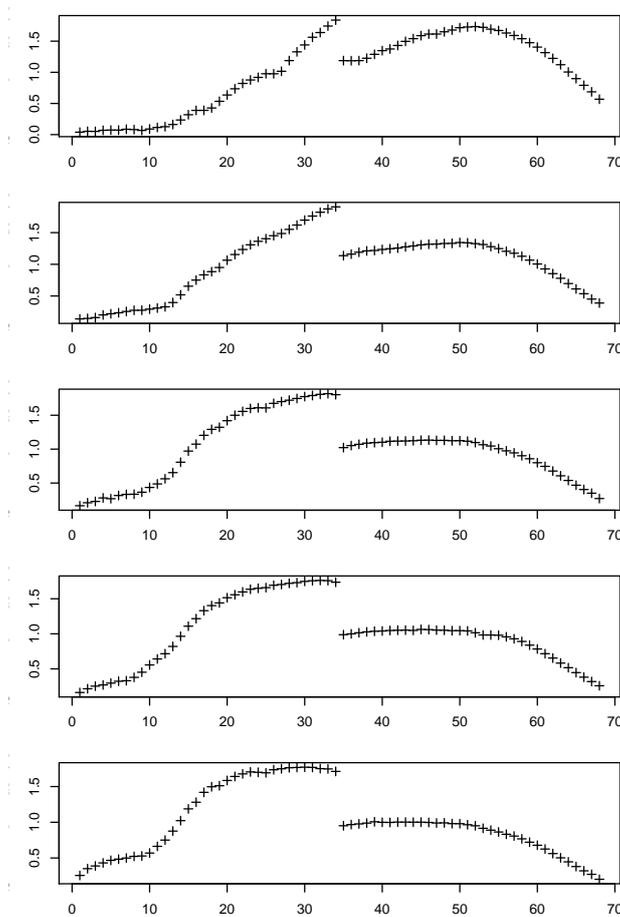


FIGURE 12: SNR weight (equation 1) for the five stars with AP evolution shown in Fig. 11 and AP spectrum updates at iteration 2 shown in Fig. 13. The mean SNR is 231–235 in the five cases.

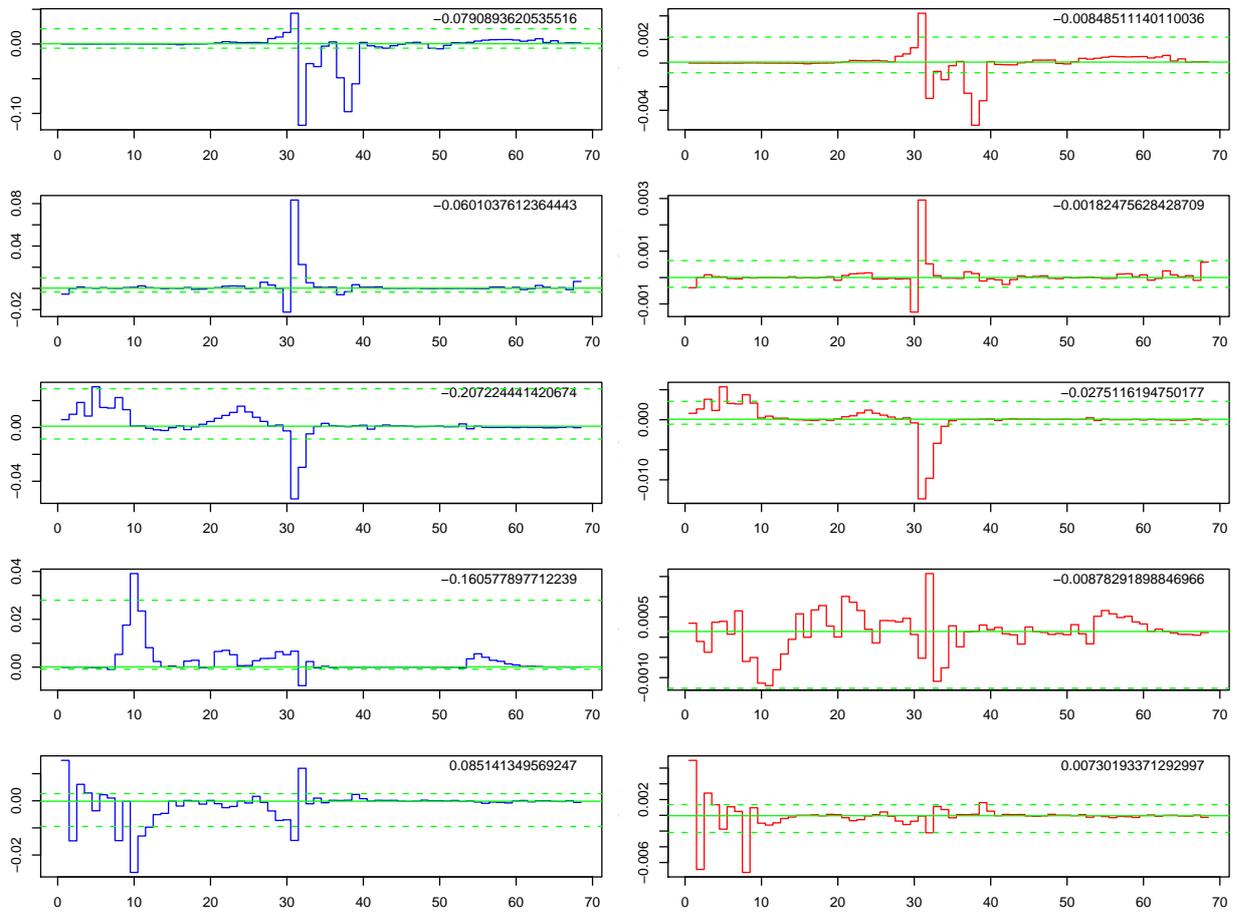


FIGURE 13: The spectrum of $\log g$ updates (blue, left column) and $\log(T_{\text{eff}})$ updates (red, right column) at the second iteration for five stars (rows). These are the same five stars shown in Figure 8 of CBJ-042. In each panel the horizontal axis is the band (pixel) number and the vertical axis the AP update, in standardized units. The solid horizontal line is the mean, and the dashed lines the upper and lower clipping limits.

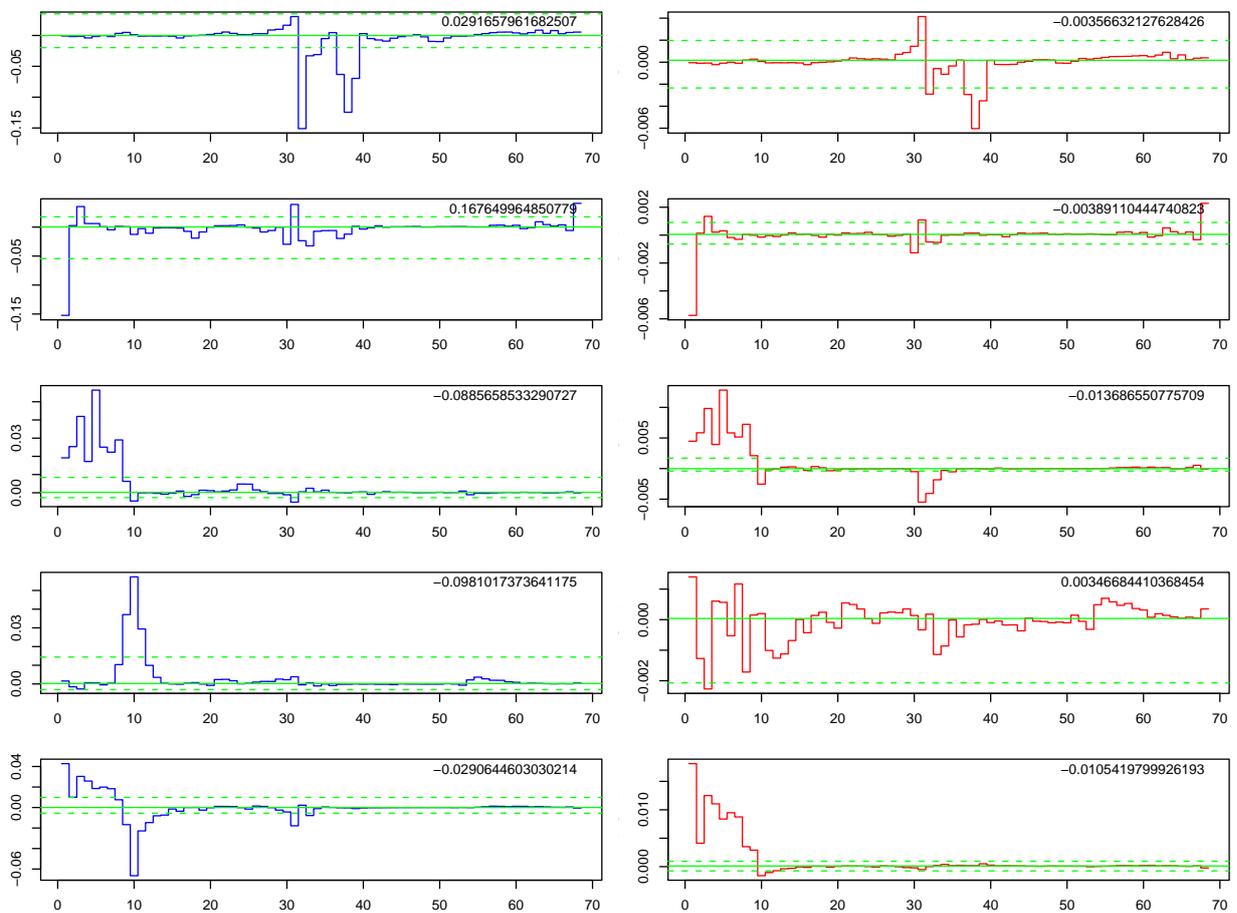


FIGURE 14: As Fig. 13 but without SNR weighting.

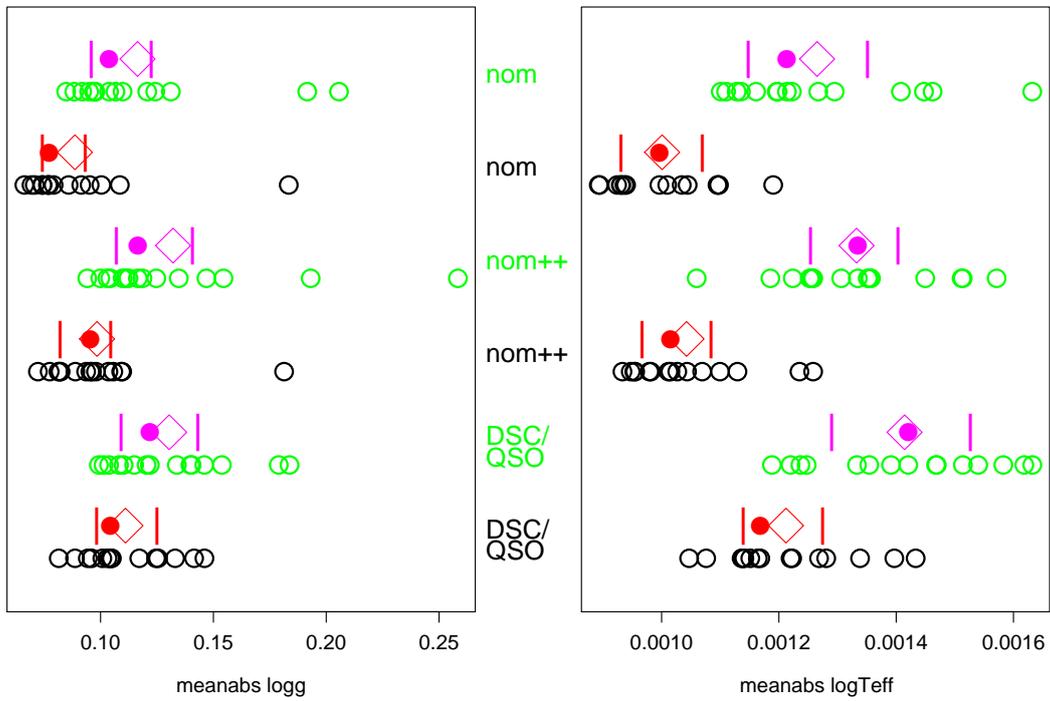


FIGURE 15: Effect of SNR weighting. Each line shows the ILIUM performance (mean absolute residual) on $T_{\text{eff}}-\log g$ problem at $G=15$ at $[\text{Fe}/\text{H}]=0.0$ using the three different input pixel ranges “nom”, “nom++”, and “DSC/QSO”, for SNR weighting included (rows 1,3,5 – the upper of each pair; points in green and magenta) and the standard ILIUM algorithm (rows 2,4,6 – the lower of each pair; points in black and red). As before, open circles are the result of different randomly selected train/test sets and the mean, median and upper/lower quartiles are plotted as a diamond, filled circle and vertical bars respectively.

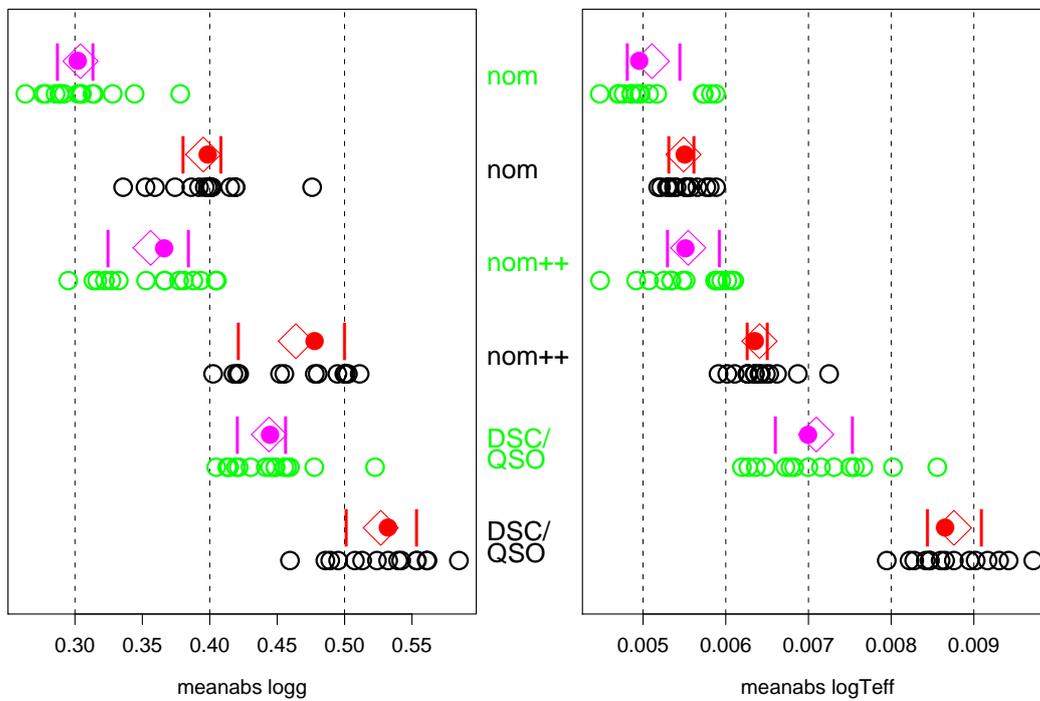


FIGURE 16: As Fig. 15 but now for G=18.5

The results are shown in Fig. 15 for $G=15$ (red/black for the standard update algorithm, magenta/green for the SNR weighted updating). Surprisingly, SNR weighting is worse than no weighting for each wavelength range, by an amount which is statistically significant for T_{eff} but not for $\log g$. We observe that the mean/median performance (magenta diamond/circle) with the SNR weighting is the same for all three wavelength ranges (the interquartile ranges overlap) for both APs. So while SNR weighting seems to degrade performance slightly, it has fulfilled the objective of making the performance independent of the wavelength range. (On the other hand, this is also true for $\log g$ without SNR weighting.)

The same examination at $G=18.5$ is shown in Fig. 16. Now the SNR weighting does decrease the mean absolute residual, by 0.1 dex for $\log g$ (up to 30%) and almost 0.002 in $\log(T_{\text{eff}})$ (up to 20%). However, we now see a dependence of the performance with wavelength coverage. As we go down the figure more noisy pixels are included and the performance gets worse, both with and without SNR weighting. So at $G=18.5$ SNR weighting helps on both parameters, but it does not eliminate a degrading effect of including noisy pixels.

In conclusion, SNR weighting has an effect, but this particular scheme does not always give the desired effect. Perhaps a weighting linear in the SNR is not optimal. We saw in the specific cases examined above that it resulted in a lot of weight being given to band 30. However, because such large updates will then be clipped (see section 2.7.4 of CBJ-042) this may actually result in the small updates being given more comparative weight after all. I therefore repeated the tests at $G=15$ with SNR weighting but without doing the clipping. But this did not improve performance (in fact it made it considerably worse in all cases). SNR weighting requires further investigation.

5.2 AP covariance estimates based on the photometric uncertainties

5.2.1 Background material

The covariance matrix between two vectors \mathbf{x} and \mathbf{y} is

$$\text{Cov}(\mathbf{x}, \mathbf{y}) = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{y} - E[\mathbf{y}])^T] \quad (4)$$

where E is the expectation operator. The covariance matrix of a single vector \mathbf{y} is

$$C_y \equiv \text{Cov}(\mathbf{y}, \mathbf{y}) = \text{Var}(\mathbf{y}) = E[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^T] \quad (5)$$

Let two vectors, \mathbf{x} and \mathbf{y} , be related by the linear transformation

$$\mathbf{y} = A\mathbf{x} \quad (6)$$

The covariance matrix of \mathbf{y} is related to the covariance matrix of \mathbf{x} by

$$\begin{aligned}
 C_y &= E[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^T] \\
 &= E[A(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T A^T] \\
 &= A E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T] A^T \\
 &= A C_x A^T
 \end{aligned} \tag{7}$$

where I assume that A is fixed, i.e. has zero variance (so $E[A\mathbf{x}] = AE[\mathbf{x}]$). The solution to the above linear equation (provided $A^T A$ is non-singular) is

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{y} \tag{8}$$

so by analogy to equation 7, we can write

$$\begin{aligned}
 C_x &= (A^T A)^{-1} A^T C_y ((A^T A)^{-1} A^T)^T \\
 &= (A^T A)^{-1} A^T C_y A (A^T A)^{-1}
 \end{aligned} \tag{9}$$

where I have used the fact that $A^T A$ and therefore $(A^T A)^{-1}$ are symmetric matrices. This is the formula for converting the covariance (errors) on \mathbf{y} to the covariance on x . If the errors on the components of y are uncorrelated and equal to σ_y^2 , then we can write $C_y = \sigma_y^2 I$ where I is the identity matrix, and equation 9 can be simplified to

$$C_x = \sigma_y^2 (A^T A)^{-1} \tag{10}$$

Another standard result from matrix algebra (derivable from equation 4 with a little algebra) is

$$\begin{aligned}
 \text{Var}(\mathbf{x} - \mathbf{y}) &= \text{Cov}(\mathbf{x}, \mathbf{x}) + \text{Cov}(\mathbf{y}, \mathbf{y}) - 2\text{Cov}(\mathbf{x}, \mathbf{y}) \\
 &= \text{Var}(\mathbf{x}) + \text{Var}(\mathbf{y}) - 2\text{Cov}(\mathbf{x}, \mathbf{y})
 \end{aligned} \tag{11}$$

5.2.2 Application to ILIUM

We can use the above results to estimate the covariance in the APs estimated by ILIUM, C_ϕ , given the covariance (measurement error) in the photometry, C_p . The basic equation of ILIUM is equation 3 in CBJ-042, which assumes that changes in the photometry can be approximated by a first order Taylor expansion using the sensitivities, i.e.

$$\delta \mathbf{p} = S \delta \phi \tag{12}$$

If we assume that ILIUM gives unbiased estimates of the APs, then we can interpret this as an expansion equation for the estimated photometry, \mathbf{p}_e , about the true photometry, \mathbf{p}_t (i.e. at the true values of the APs, ϕ_t), given the estimated APs, ϕ_e . We can then write the above equation as

$$\mathbf{p}_e - \mathbf{p}_t = S(\phi_e - \phi_t) \tag{13}$$

where S is calculated at ϕ_t in both cases. Using equation 7 we see that the covariances are related as

$$\text{Var}(\mathbf{p}_e - \mathbf{p}_t) = S \text{Var}(\phi_e - \phi_t) S^T \quad (14)$$

Using equation 11 we can write this as

$$\text{Var}(\mathbf{p}_e) + \text{Var}(\mathbf{p}_t) - 2\text{Cov}(\mathbf{p}_e, \mathbf{p}_t) = S (\text{Var}(\phi_e) + \text{Var}(\phi_t) - 2\text{Cov}(\phi_e, \phi_t)) S^T \quad (15)$$

However, as ϕ_t and \mathbf{p}_t are the true values, they have zero variance. Moreover, as we assume that ILIUM is giving unbiased estimates, these true values are also uncorrelated with the estimated values (i.e. the estimates are scattered randomly around them). Therefore, the Cov terms in the above equation are also zero and it becomes

$$\begin{aligned} \text{Var}(\mathbf{p}_e) &= S \text{Var}(\phi_e) S^T \\ C_p &= S C_\phi S^T \end{aligned} \quad (16)$$

Using equation 9 we get

$$C_\phi = (S^T S)^{-1} S^T C_p S (S^T S)^{-1} \quad (17)$$

Thus given the estimated variance (more generally, covariance) in the measured photometry we can calculate the corresponding covariance in the APs using the sensitivity at the estimated APs. C_ϕ is a symmetric $J \times J$ matrix with elements $c_{jj'}$. The AP variances are the diagonal elements, so the estimated uncertainties on the APs are the square roots of these, $\sqrt{\text{diag}(C_\phi)}$, which write as ϵ_{feh} and ϵ_{teff} (where the latter always refers to $\log(T_{\text{eff}})$). Rather than reporting the off-diagonal covariances, I will report the correlation coefficient between two APs j and j' , $r_{jj'} = c_{jj'} / \sqrt{c_{jj} c_{j'j'}}$, which, as there are just two APs, I write as r . (I reserve the symbols ρ and σ for statistical measures of the correlation and standard deviation of the residuals.)

Strictly speaking we should use the sensitivity at the (unknown) *true* APs in equation 17. The above is an approximation which will be reasonably valid if the AP error is not large. Furthermore, it will hold on average over the sample if ILIUM gives unbiased estimates, which seems to be the case given the very low systematic errors I have reported in CBJ-042 and CBJ-043. Equation 17 also assumes that there are no errors in S , which in turn assumes the forward model is perfect. We could accommodate errors in the sensitivities by generalizing equation 7 to include the covariance in A . If the photometric errors are uncorrelated, then we can write $C_p = \text{diag}(\sigma_p^2)$, where σ_p is the vector of estimated errors in each band. (In general C_ϕ will still have non-zero off-diagonal elements.) If we had no individual error estimates on the photometric observations, then we could use equation 10 and set σ to some overall estimate of the noise in \mathbf{p} .

For computational reasons we can calculate equation 17 using

$$C_\phi = M C_p M^T \quad (18)$$

where

$$M = (S^T S)^{-1} S^T \quad (19)$$

It will be recalled from section 2.7.4 of CBJ-042 that the update equation in ILIUM is just $\delta\phi = -M \delta p$. There I also defined a different estimate of the AP uncertainties, namely $|M \delta p|$, which is based on the difference between the measured and predicted photometry. Here we use an estimate based on the uncertainties in the photometry.

It is worth noting that there are some similarities between the principles of ILIUM and the figure-of-merit defined to assess the performance of photometric systems in the earlier Gaia filter design design process (see Lindegren et al. 2003 and Jordi et al. 2006). Sensitivities were also used by Bailer-Jones (2003, 2004) in the Heuristic Filter Design approach.

5.2.3 Results

I used the above to estimate the AP uncertainties and correlations for the T_{eff} -[Fe/H] problem for dwarfs at $G=20$ (section 2.3 of CBJ-043). Fig. 17 shows various distributions and scatter plots. What is not visible in these is the fact that a very large fraction of objects have large estimated uncertainties, especially in [Fe/H] where they can reach tens of dex. Some $282/858 = 33\%$ of objects have an uncertainty ratio (estimated error divided by modulus of true residual) in [Fe/H] greater than 8 (and so are not plotted in the top row). This reduces to $54/338 = 16\%$ when we limit the sample to cool stars ($T_{\text{eff}} \leq 7000$ K). For the T_{eff} uncertainty ratio it is 12% for both the full and cool star sample. This is a considerably larger fraction than was obtained with the original uncertainty estimate technique (section 2.6 of CBJ-042), which gave just 7% of objects beyond this level. Comparing the top two panels of Fig. 17 with those in Fig. 12 of CBJ-043 also shows that the ratios are generally higher (i.e. overestimated) with the new method. Fig. 17 also shows the actual values of the estimated uncertainties as a function of AP. Larger uncertainties in [Fe/H] are predicted for higher T_{eff} (as expected), and what is also interesting is that the [Fe/H] uncertainty is never predicted to be small for larger T_{eff} , which is a correct reflection of the situation. Interestingly, the uncertainty in $\log(T_{\text{eff}})$ is also larger at higher T_{eff} . This may be a result of the larger [Fe/H] uncertainty and a possible correlation between them. Indeed, the right-hand panel in the second row shows that the correlations can be very large. On the other hand, the original method gave many error estimates which were significant *underestimates* of the residuals (the peak around 0 in the histograms of Figure 12 of CBJ-043), something which does not occur here so is in favour of the new method.

Fig. 18 plots the same estimates but now on the grid of the true APs. It illustrates more clearly the growth of errors for large T_{eff} . The bottom panel plots the size and magnitude of the estimated correlations. Negative correlations are predicted for hot stars and low metallicity stars, that is, an overestimation of one AP is predicted to accompany an underestimation of the other. We can compare this with the plot of the magnitudes of the residuals (i.e. estimated minus true) shown in the top two panels of Fig. 19, where I have used the same scale for the symbols. The [Fe/H] uncertainties are significant overestimates of the [Fe/H] residuals, whereas the T_{eff} uncertainties are also overestimated, but not by so much. The bottom panel shows the sign of the product of the two residuals, i.e. the sign of the correlation. This does not show the same trend

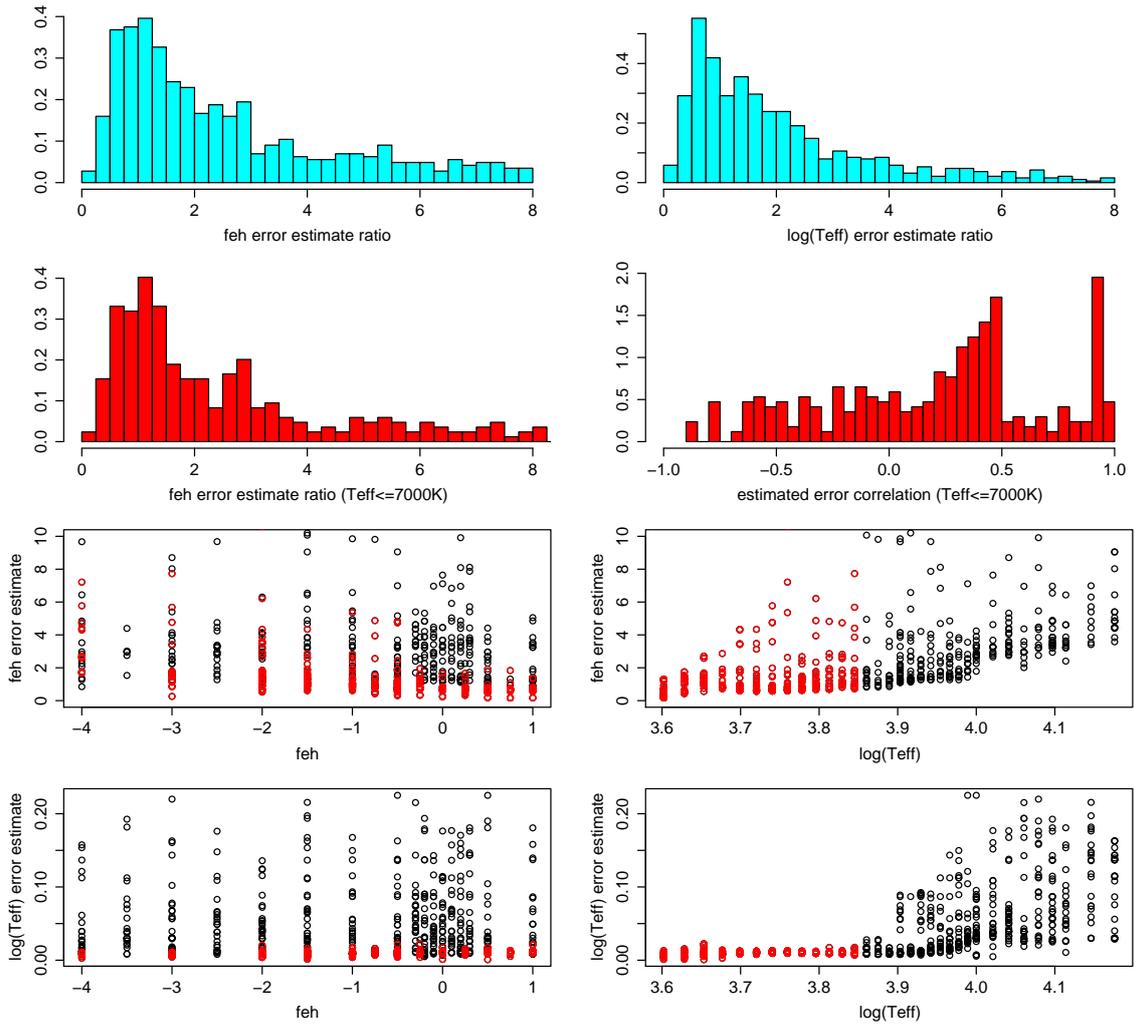


FIGURE 17: Predicted AP uncertainties and correlations for the $T_{\text{eff}}-[Fe/H]$ problem for dwarfs at $G=20$. Top row: AP uncertainties expressed as a ratio of the absolute value of the true residuals. Second row, left: as above, but only for cool stars ($T_{\text{eff}} \leq 7000$ K). Second row, right: the estimated correlations, r , only for cool stars. Third and fourth rows: estimated uncertainties (not ratios) as a function of the APs, with cool stars plotted in red.

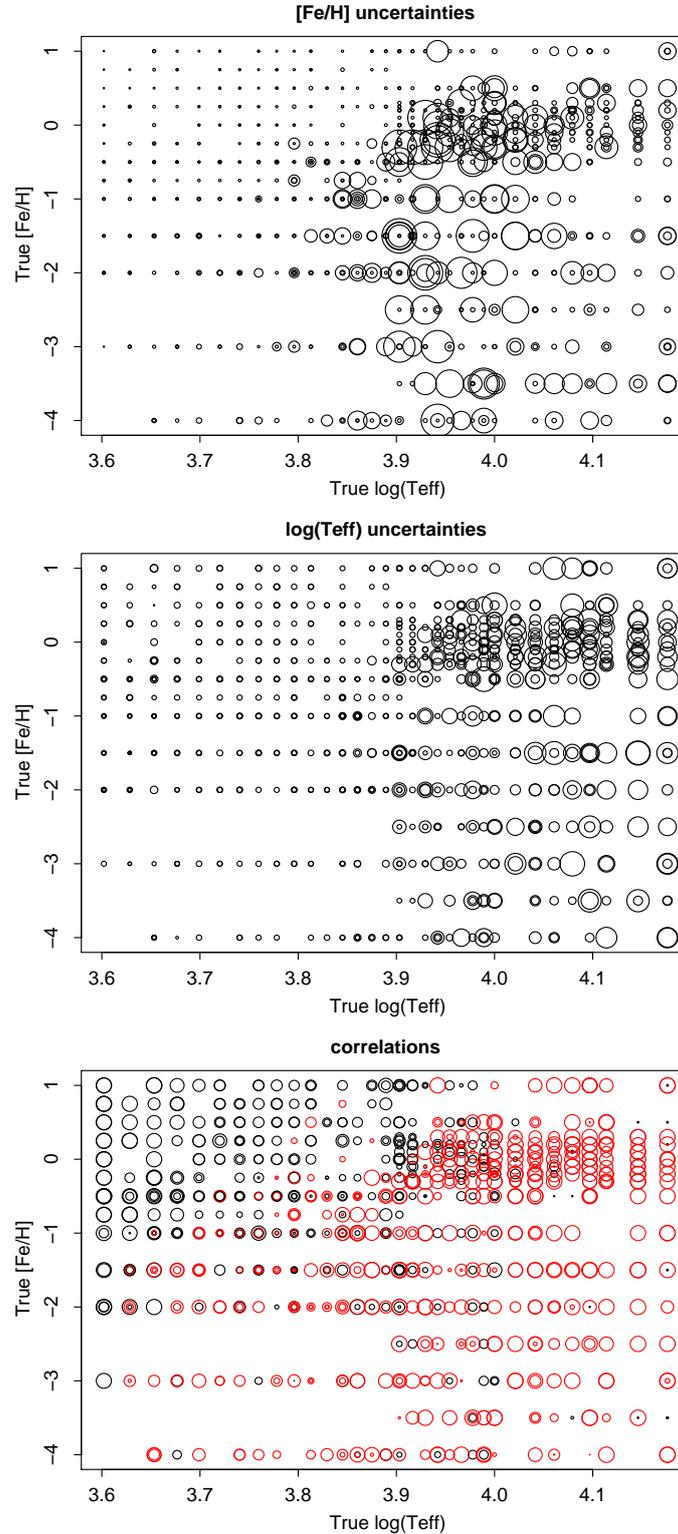


FIGURE 18: Predicted AP uncertainties and correlations for the $T_{\text{eff}}-[Fe/H]$ problem for dwarfs at $G=20$ for individual objects plotted in the space of the true APs. The area of the plotted circle (not the diameter) is proportional to the size of the uncertainty or correlation. For the latter, positive values are shown in black and negative in red. (Not all points are present because the R plotting software does not plot any symbol below a certain size.)

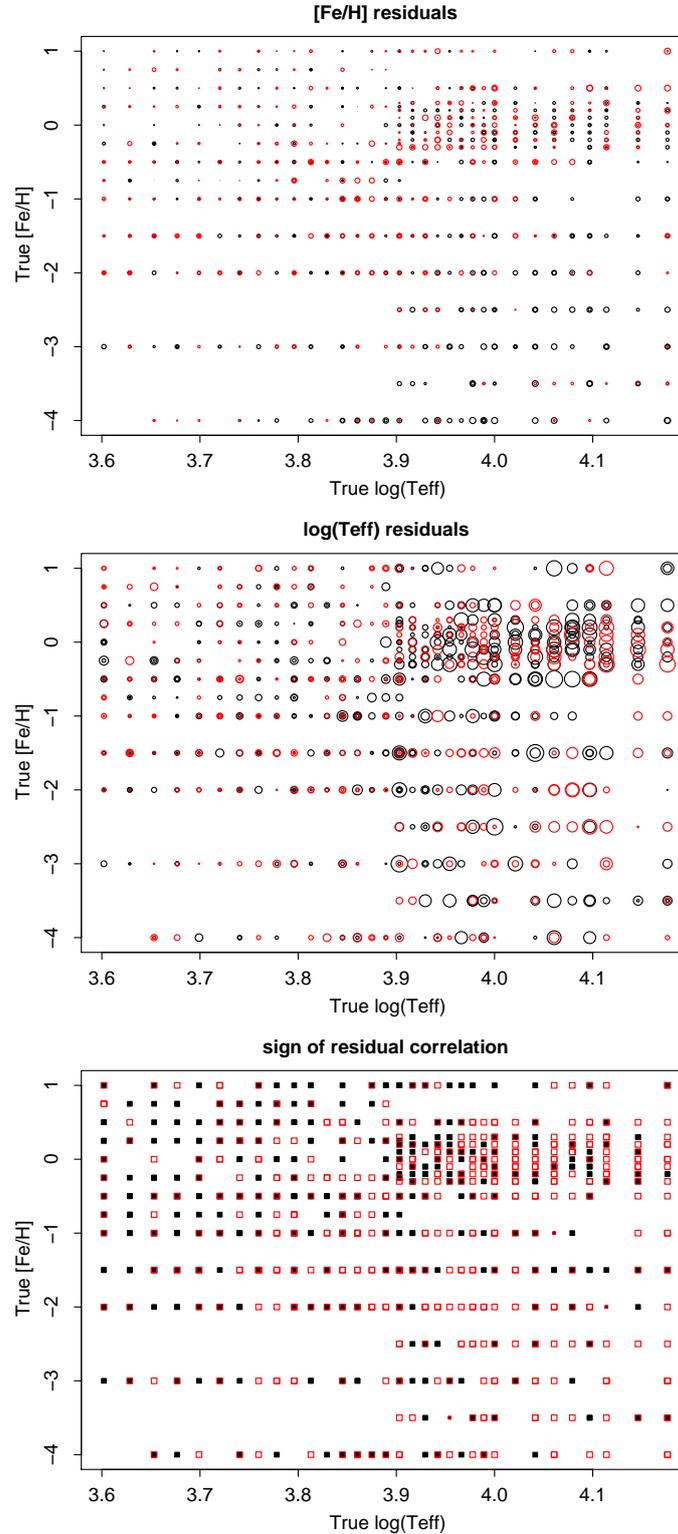


FIGURE 19: T_{eff} residuals (top) and $[\text{Fe}/\text{H}]$ residuals (middle) for the $T_{\text{eff}}-[\text{Fe}/\text{H}]$ problem for dwarfs at $G=20$. The area of the plotted circle (not the diameter) is proportional to the size of the residual and the colour the sign (black positive, red negative). The bottom panel shows the sign of the product of the residuals: filled black points indicate a positive correlation and open red squares a negative one.

to negative correlations for hot and low metallicity stars we see predicted in Fig. 18.

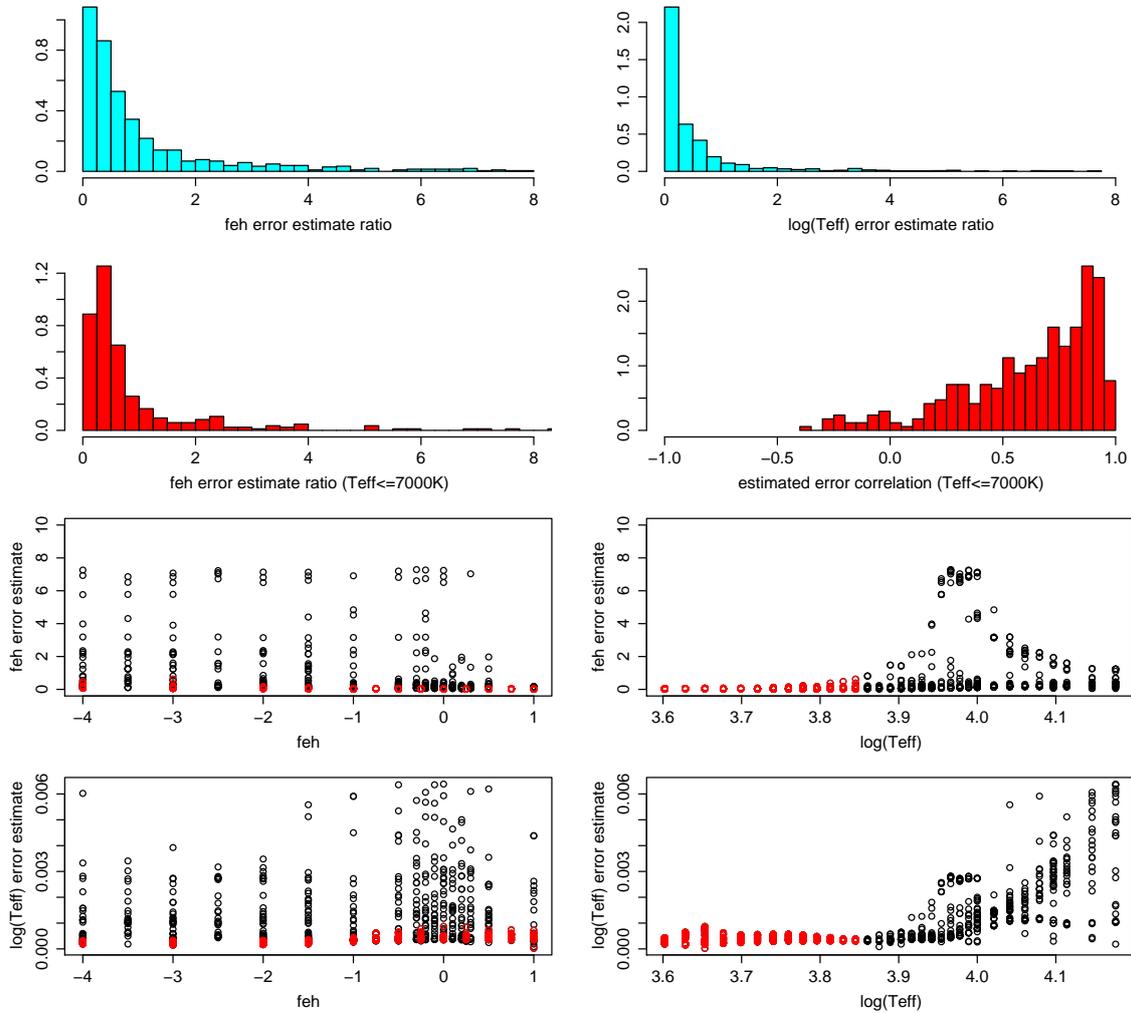


FIGURE 20: As Fig. 17 but for G=15

Figs. 20, 21, 22 show the same three diagnostic plots but now for the data at G=15 (section 2.1 of CBJ-043). As the residuals and the uncertainties are now much smaller, I have increased the area of the points by a factor of 16. The uncertainty estimates are closer to the true residuals and not nearly as overestimated as was the case at G=20. The histograms show that the uncertainties expressed as a fraction of the residuals are much smaller. In fact, they now have a tendency to be underestimated, as was also seen at G=15 with the original uncertainty estimation method (Figure 9 of CBJ-043). The agreement of the predicted and actual correlations in the uncertainties/residuals is perhaps better than at G=20.

This uncertainty estimation method has a more solid grounding than the rather naive one de-

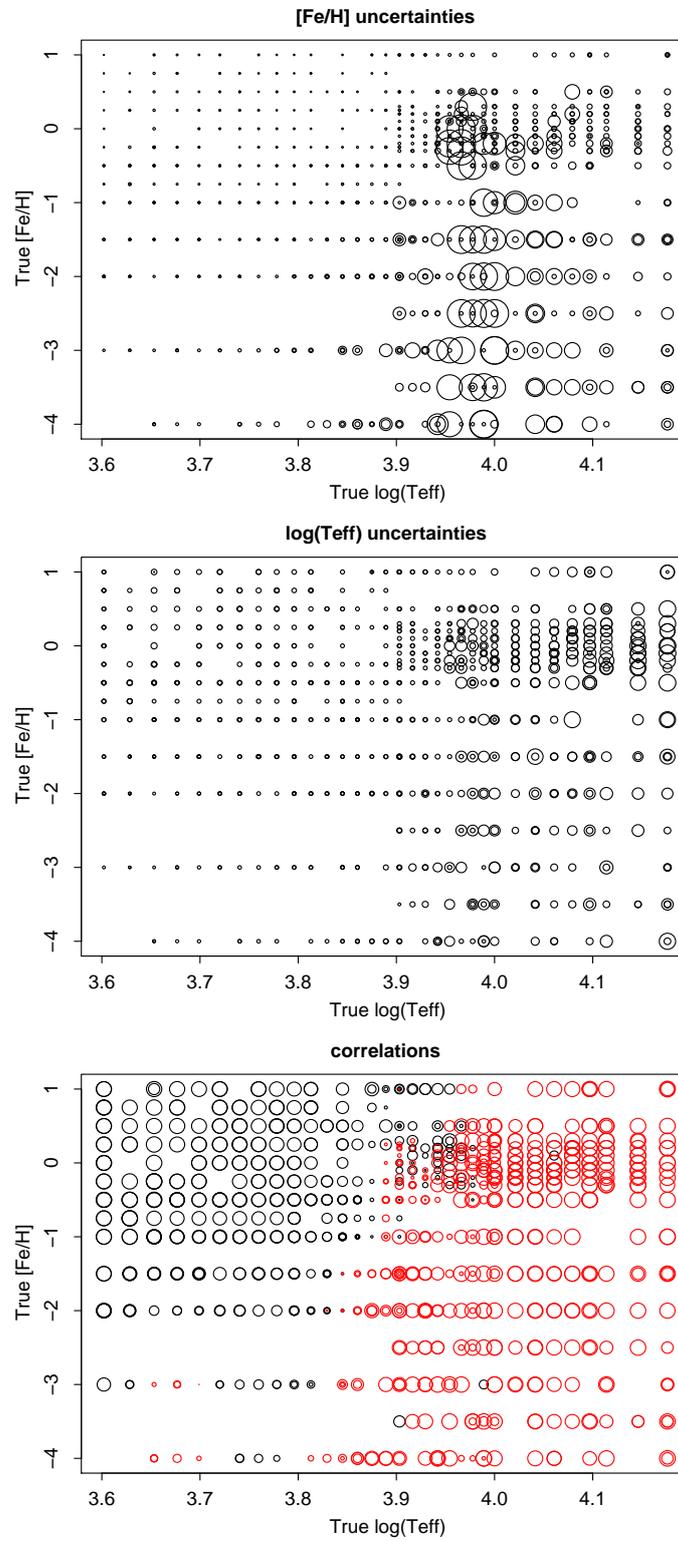


FIGURE 21: As Fig. 18 but for G=15 and with the circles plotted 16 times larger in area.

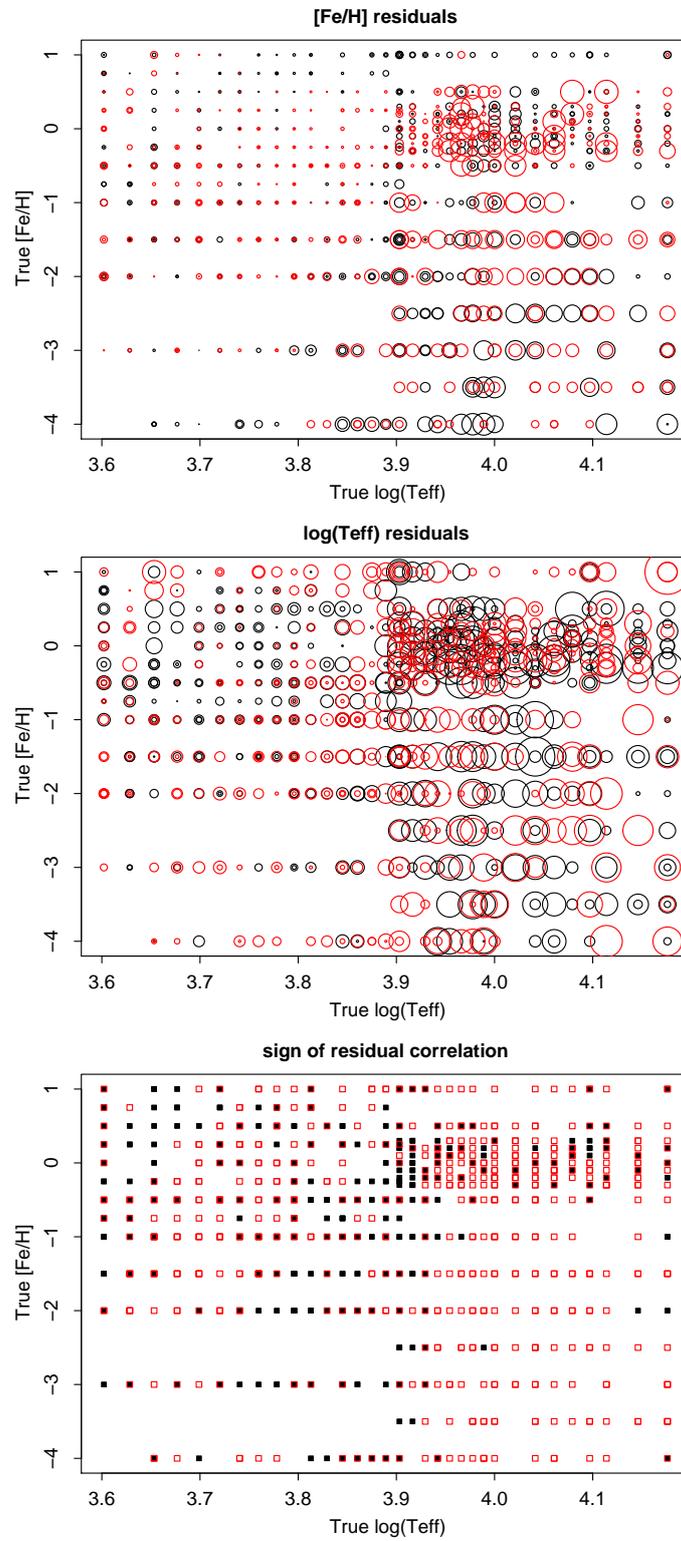


FIGURE 22: As Fig. 19 but for G=15 and with the circles plotted 16 times larger in area.

scribed and used in CBJ-042 and CBJ-043. The new method depends explicitly on the estimated photometric error rather than a sample of the AP update oscillation. Yet based on this one example, its not clear that the new method gives more accurate uncertainty estimates (it seems worse at $G=20$, perhaps similar in performance at $G=15$). A more rigorous assessment is required.

References

Bailer-Jones C.A.L., 2003, *Heuristic design of filter systems using an evolutionary algorithm* Presentation at PWG meeting, 9 October 2003, Leiden

Bailer-Jones C.A.L., 2004, *Evolutionary design of photometric systems and its application to Gaia*, A&A 419, 385

Bailer-Jones C.A.L., 2009, *ILIUM: An iterative local interpolation method for parameter estimation*, GAIA-C8-TN-MPIA-CBJ-042

Bailer-Jones C.A.L., 2009, *Application of ILIUM to the estimation of the $T_{\text{eff}}-[Fe/H]$ pair from BP/RP*, GAIA-C8-TN-MPIA-CBJ-043

Bailer-Jones C.A.L., 2009, *A procedure to calibrate algorithms for estimating parameters from spectra*, GAIA-C8-TN-MPIA-CBJ-044

Bailer-Jones C.A.L., Smith K.W., Tiede C., Sordo R., Vallenari A., 2008, *Finding rare objects and building pure samples: probabilistic quasar classification from low-resolution Gaia spectra*, MNRAS 391, 1838

Jordi C., Hog E., Brown A.G.A., Lindegren L., Bailer-Jones C.A.L., Carrasco J.M., Knude J., Straizys V., et al., 2006, *The design and performance of the Gaia photometric system* MNRAS 367, 290

Lindegren, L., 2003, *Optimizing Gaia's photometric system - thoughts on distance measure and figure of merit Functions*, GAIA-LL-047

Lopez Marti B., Figueras F., Jordi C., Carrasco J.M., Gebran M., 2009, *Source with spectral features as seen by Gaia BP/RP* GAIA-C5-UB-BLM-001-1