$\Lambda_c^+ \to phh'$ yield stabilities and PID

Stephen Ogilvy

July 21, 2015

This document contains a series of cross checks pertaining to the prompt $\Lambda_c^+ \to phh'$ BR analysis. We focus mainly on the ratio $\mathcal{B}(\Lambda_c^+ \to pK^-K^+)/\mathcal{B}(\Lambda_c^+ \to pK^-\pi^+)$ where we consider just the raw yields.

1 Studies with 20 % of data

These studies are carried out with a subset of the data roughly equal to 20 % of the stripping output for each mode. These are selected as follows:

Stripping17b - The full 17b dataset is processed by a DaVinci job. All subjob outputs are merged into a TChain and the first 20 % of these events is the dataset used, as rounded by an integer from float python comparison, so nearest integer lower than the float value.

Stripping21r1 - The first 20 % of the MDST files is taken as an input to the DaVinci job, with the 20 % calculated as for the Stripping17b output.

So we expect the integrated luminosities of the stripping versions to be not quite the same in this study. Accordingly the ratios of MagUp to MagDown should be different between the strippings. Will eventually give more info on the fits in Section A, but for now make do with saying all fits were convergent with accurate error matrices.

1.1 Definitions of selections

The selections herein are the stripping selections, the TIS trigger requirement on the Λ_c^+ candidate, plus additional PID cuts. In Stripping21r1 where the maximum Λ_c lifetime cut is not present, I've added the maximum Λ_c lifetime cut of 1.2 ps from Stripping17b to make the selections identical.

In this document we use three different additional PID selections for sanity checks - loose, moderate and tight. These are defined in Table 1. I emphasise that these cuts apply to all particles in the Λ_c decay - so in the case of $\Lambda_c^+ \to pK^-K^+$ the kaon cuts apply to both kaons.

selection	$p\ \log(\mathcal{L}_p/\mathcal{L}_K)$	$p\ \log(\mathcal{L}_p/\mathcal{L}_\pi)$	$K\ \log(\mathcal{L}_K/\mathcal{L}_p)$	$K\ \log(\mathcal{L}_K/\mathcal{L}_\pi)$
loose	5	10	-12	8
moderate	8	15	-8	12
tight	11	21	-6	16

Table 1: The PID selection definitions. The values given are the minumum DLL required for the track to be accepted. Selection applies to all particles of that species in the Λ_c decay.

1.2 Stability of polarities

The first cross check is more of a sanity check that all the job processing and scripts are working correctly - are the raw yield ratios of MagUp and MagDown consistent? We first define the ratio:

$$R_P = \frac{N(\Lambda_c^+ \to pK^-\pi^+)\ \text{MagDown}}{N(\Lambda_c^+ \to pK^-\pi^+)\ \text{MagUp}} \tag{1}$$

where N is the total Λ_c^+ yield for that decay mode, as established from a fit to the Λ_c^+ mass. Therefore the yield is the union of the prompt and the secondary Λ_c^+ in the sample.

We give these ratios for $\Lambda_c^+ \to pK^-\pi^+$ for the highest precision of the $\Lambda_c^+ \to phh'$ modes. I also found it impossible to extract significant $\Lambda_c^+ \to pK^-K^+$ signals for this subset of the data using the loose selections, so I've done this for just the Cabibbo-favoured mode. The results are presented in Table 2. The ratios are consistent within the individual stripping processings.

Stripping	PID selection	R_P
17b	loose	1.49 ± 0.03
	moderate	1.49 ± 0.03
	tight	1.50 ± 0.03
21r1	loose	1.42 ± 0.03
	moderate	1.42 ± 0.03
	tight	1.44 ± 0.03

Table 2: The values of R_P for each stripping version and PID selection. The ratios are consistent.

1.3 Stability of CS/CF modes

Now I wanted to look at the yields of the $N(\Lambda_c^+ \to pK^-K^+)/N(\Lambda_c^+ \to pK^-\pi^+)$, to check they were consistent between MagUp and MagDown. Define these ratios of raw signal yields as R_{up} and R_{down} for MagUp and MagDown respectively. Statistically significant signals for the $\Lambda_c^+ \to pK^-K^+$ mode can be derived for the moderate and tight PID selections for this sample of the data, but not for the loose PID selections. As such we present these ratios for only the moderate and tight PID selections. They are given in Table 3.

Selection	Stripping	R_{down} [%]	R_{up} [%]
Moderate	17b	1.34 ± 0.20	1.24 ± 0.23
	21r1	1.42 ± 0.21	1.22 ± 0.23
Tight	17b	1.83 ± 0.28	1.69 ± 0.21
	21r1	1.30 ± 0.21	1.57 ± 0.23

Table 3: The values of R_{up} and R_{down} for the each stripping, polarity, and PID selection.

Several things are noticeable here - in the moderate PID selection, the ratios of up and down agree within errors, as do the various stripping versions. Everything seems consistent there. When we go to the tight selection, two things are immediately obvious:

- In the tight selection, there is a marked disagreement between MagUp and MagDown in Stripping21r1.
- R_{down} disagrees strongly between Stripping17b and Stripping21r1.
- The most striking observation to be made - *when we tighten the cuts on kaons, the ratio* $N(\Lambda_c^+ \to pK^-K^+)/N(\Lambda_c^+ \to pK^-\pi^+)$ *increases in all cases.* Three of the four tight selection samples display statistically significant increases relative to the equivalent moderate selection values.

Naively, we would expect this ratio to be reduced the tighter we cut on kaons, by virtue of there being two kaons in the suppressed final state and one in the favoured final state. There are no tighter PID requirements being made on the pion in any of these selections.

We would like to see some more info here, and the first step is by repeating the studies using the full dataset so we have enough precision to analyse the signal components with the ${}_s\mathcal{P}lots$ technique. We can also see if these discrepancies hold up with greater precision.

2 Studies with full dataset

So now I want to measure the ratios R_{up} and R_{down} for the full datasets to get greater precision on the results, so verify these are significant differences. So we do this with the identical selections described previously. They are given in Table 4.

Selection	Stripping	R_{down} [%]	R_{up} [%]
Moderate	17b	1.91 ± 0.11	1.84 ± 0.12
	21r1	1.58 ± 0.14	1.34 ± 0.14
Tight	17b	1.49 ± 0.10	1.63 ± 0.09
	21r1	1.40 ± 0.11	1.62 ± 0.11

Table 4: The values of R_{up} and R_{down} for the each stripping, polarity, and PID selection.

So now, for the full dataset, we begin to observe some of the expected beahviour with Stripping17b, although the ratios in 17b still seem suspiciously high to me. Why are these so far from the subsets I ran? I'm also worried how the only section where MagUp and MagDown are compatible are in 17b. In all other cases the raw yields are not stable with respect to magnet polarity. Something very odd is happening here - is it to do with the data processing? Or is this to do with the fit itself?

2.1 Including fits to combined polarities

Let's compare the individual polarity fits to the combined fit. Let's call the total ratio R_{total}. Now the full results are replicated with the total ratios in Table 5.

Selection	Stripping	R_{down} [%]	R_{up} [%]	R_{total} [%]
Moderate	17b	1.91 ± 0.11	1.84 ± 0.12	1.89 ± 0.08
	21r1	1.58 ± 0.14	1.34 ± 0.14	1.47 ± 0.10
Tight	17b	1.49 ± 0.10	1.63 ± 0.09	1.57 ± 0.07
	21r1	1.40 ± 0.11	1.62 ± 0.11	1.52 ± 0.08

Table 5: The values of R_{up}, R_{down}, R_{total} for the each stripping, polarity, and PID selection.

Now let's check the totals are compatible between the combined fit signal yield and the individual polarity fits. These are given in Table 6.

Selection	Stripping	Mode	N_{down}	N_{up}	$N_{\mathrm{up}} + N_{\mathrm{down}}$	N_{total}
Moderate	17b	$\Lambda_c^+ \to pK^-K^+$	1783 ± 103	1381 ± 89	3164 ± 136	3174 ± 137
		$\Lambda_c^+ \to pK^-\pi^+$	93149 ± 500	74951 ± 463	168100 ± 681	168088 ± 696
	21r1	$\Lambda_c^+ \to pK^-K^+$	1530 ± 134	944 ± 99	2474 ± 167	2456 ± 165
		$\Lambda_c^+ \to pK^-\pi^+$	96622 ± 465	70288 ± 520	166910 ± 698	167054 ± 842
Tight	17b	$\Lambda_c^+ \to pK^-K^+$	1009 ± 58	743 ± 50	1752 ± 82	1750 ± 77
		$\Lambda_c^+ \to pK^-\pi^+$	61736 ± 345	49891 ± 251	111627 ± 427	11597 ± 478
	21r1	$\Lambda_c^+ \to pK^-K^+$	1039 ± 68	662 ± 54	1701 ± 87	1699 ± 87
		$\Lambda_c^+ \to pK^-\pi^+$	64176 ± 371	47231 ± 303	111407 ± 728	111450 ± 549

Table 6: The yields from the fits by polarity and the fit to the union of the polarities.

These all look quite convincing - the fit seems stable enough between the statistics of the individual polarity samples and the statistics of the full samples. This still raises the qestion - why are the ratios of R_{down} and R_{up} so different in Stripping21r1?

3 Summary of outstanding questions

So a quick recap of some questions these studies highlight for me:

1. Why are the MagUp and Magdown raw ratios so different in Stripping21r1?
2. Why does the total raw ratio agree between the stripping versions for the tight PID selection, but is much lower for the moderate?

Considering the second point - the shape of the combinatoric isn't very well modelled in the moderate fits. There's going to be some reflection component under there from $D^+ \to hhh$ decays, which is especially obvious in the high mass region where the combinatoric tails off for $\Lambda_c^+ \to pK^-K^+$.

Still, it doesn't seem that off, and I wouldn't expect this to drastically affect the signal yields in $\Lambda_c^+ \to pK^-K^+$ given how narrow the signal peak is - it's difficult for some broad underlying feature to drastically affect the raw extracted yields. If we have to use signal weighting, there could be some problems with reflections under the combinatoric - but these are only very obvious in the moderate PID selection which we don't want to use for the full analysis anyway.

Another peculiarity is that the signal width in the moderate 21r1 $\Lambda_c^+ \to pK^-K^+$ is markedly lower than the other fits. Usually the width of the peak reported by the fit is around 4.2 MeV, while here it's significantly lower at (3.4 ± 0.3) MeV. I can't immediately think of a cause for this - I wouldn't expect broad refelctions so significantly affect such a narrow peak but there aren't many other explanations for why the peak is reported as narrower than usual.

4 How on earth can I address these questions

Ok - we're getting some bizarre effects. I'm not immediately sure how to answer them. Next up it might be good to run the PID efficiencies for the selections in our data-driven fashion. For Stripping21r1 this is no problem, just use the most recent `PIDCalib`. For 17b this'll be a pain - the old datasets may still be somewhere eaasily accessible and it's possible that an older version of PIDCalib can still make Perfhists easily. If not I'll have to make a script to generate the perfhists myself - it's a simple enough procedure but enough of a pain given the calibrations sample's implementation as a RooDataSet. Will do the Stripping21r1 efficiencies first.

I also want to take a look at the raw ratios once the rest of the offline cuts are employed - so once we also add the kinematic vetoes and DTF convergence criterion. Can do these both independently of one another.

5 Raw yields after full selection

So now we also apply the DTF convergence requirement and the kinematic vetoes *to both Stripping21r1 and 17b.* I want the selections to be identical so I can check for any bizarre effects appearing. These are summed up in Table 7. For the time being I'm just going to go with the tight selection, as this'll be closer to the one we'll use.

Selection	Stripping	R_{down} [%]	R_{up} [%]	R_{total} [%]
Tight	17b	1.61 ± 0.11	1.53 ± 0.09	1.62 ± 0.08
	21r1	1.65 ± 0.11	1.41 ± 0.13	1.55 ± 0.09

Table 7: The values of R_{up}, R_{down}, R_{total} for the each stripping, polarity, and PID selection, with the full selection applied, including kinematic vetoes and DTF convergence requirement.

The raw ratios are broadly similar - but they are lower than those given in the analysis note. Something very fishy is going on here - there's either a problem with this dataset or the one used in the analysis note. They should be the same but maybe there's been an issue copying them over to mass storage? Need to figure this out - for now I want to verify that after correction these will be different from the current ANA figures - after all the PID selections are very slightly different. I doubt they're different enough to explain this, but let's attempt an efficiency correction using `PIDCalib` and the datasets with no kinematic vetoes.

6 Stripping21r1 PID efficiencies

This is as good a time as any to have a look at the new proton samples introduced in Jun15. These take into account the new Λ_c from Λ_b^0 samples Sneha just released. Using these, I can take a look at the PID-adjusted yields without any kinematic vetoes applied.

The samples are processed seperately, so the histograms for the proton samples need merging. There are several ways to implement this, and both are a bit of a pain. I can either make weighted average histograms or I can try to take a value from the Λ^0 sample and if there's no valid efficiency instead look in the Λ_c samples. Given how many more tracks there are in the Λ^0 samples this effectively amounts to the exact same thing.

So after some testing, there's a slight problem with this approach. The track yields come from sums of weights from the ${}_s\mathcal{P}lots$ technique, and so in some bins we have invalid efficiencies or total numbers of tracks below zero. My initial workaround was to add the total and passed number of tracks and to make new efficiency histograms, but this doesn't work very well because of these negative weights. This is severe enough to have the number of tracks in the Lc sample drowned out by the negative weights in the Lz sample - unsurprising given the sum of weights errors on the Lz sample, which corresponds to hundreds of millions of tracks. Having a the level of a few hundred on the error where no signal is present, just from the combinatoric weights, is common.

So now I've cooked up a new way of doing this. Need an algorithm that will:

1. Take as input histograms with the efficiency and the histograms for the total number of tracks for both the Λ^0 and Λ_c samples.
2. Also generate a new empty histogram with the same bin structure as the two samples.
3. On a binwise basis, check that both the efficiency and the number of tracks in a sample are positive and well defined. If both are take the weighted average of the efficiencies, and fill that value to the corresponding bin of the new histogram.
4. If only one sample has a valid bin efficiency and number of tracks, like where I expect only Λ_c tracks in the high p_{T} region, take the efficiency as the valid one and fill that to the corresponding bin in the new histogram.
5. If both are invalid, complain - the bins should be redefined such that all bins have valid tracks.

So now that's done and dusted. I've got PID weights out with coarse binning schemas, from which I'd expect systematics of the order of 3 – 5 %. So we have these for each magnet polarity. I'm thinking we do the adjusted yields for each individual polarity.

6.1 Data reference sample `PIDCalib`

It would probably be a good idea to examine the PID efficiencies using both a data reference sample and a simulated reference sample. For now we only have the former, so let's get adjusted yields using the "standard" approach in this analysis. That is, we get a PID efficiency using the ${}_s\mathcal{P}lots$ signal weights and the PIDCalib per-event efficencies. So the PID efficiency is given as:

$$\epsilon = \frac{\sum_{i=0}^{n} w_i}{\sum_{i=0}^{n} w_i \epsilon_i} \tag{2}$$

where we run over n events, w_i are the signal weights and ϵ_i is the candidate `PIDCalib` efficiency. These are given for the tight PID selection in Table 8.

Now, despite really tightening up the PID selections on the kaons here we're not seeing a huge drop off in efficiency. There could be many feasible reasons why this might be the case. Next up I want to have the PID adjusted ratios compared to the raw ratios. We'll define P_{up} and P_{down} as the ratio of $\Lambda_c^+ \to pK^-K^+$ to $\Lambda_c^+ \to pK^-\pi^+$ after PID efficiency correction. These are compared in Table 9.

The results are very different between the magnet polarities.

For now assume an error of 2 % on the PID efficiencies motivated by the difference between the MagUp and MagDown efficiencies for $\Lambda_c^+ \to pK^-\pi^+$- this can be properly evaluated later. Combining the two independent polarity results we get (1.64 ± 0.16) %, where the error takes into account the rough PID error and the error on both the CF ans CS signal yields, all summed in quadrature. This is much lower than I would have anticipated. Could be a good sign - just not sure why this is now much lower than for the previous 17b analysis.

Selection	Mode	Polarity	ϵ [%]
Tight	$\Lambda_c^+ \to pK^-\pi^+$	MagDown	25.0
		MagUp	23.1
	$\Lambda_c^+ \to pK^-K^+$	MagDown	22.5
		MagUp	22.3

Table 8: Data-driven PID efficiencies for Stripping21r1 from PIDCalib.

Selection	Stripping	R_{down} [%]	R_{up} [%]	P_{down} [%]	P_{up} [%]
Tight	21r1	1.65 ± 0.11	1.41 ± 0.13	1.83 ± 0.12	1.46 ± 0.13

Table 9: The values of R_{up}, R_{down}, R_{total} for the each stripping, polarity, and PID selection, with the full selection applied, including kinematic vetoes and DTF convergence requirement.

7 Stripping21r1 Prompt simulation

Have moved onto the task of running over the simulation to produce ntuples for the 21r1 simulation. These use the Stripping 20r1 simulation that's in the book-keeping. This means I have to run a taylored version of the 21r1 stripping on the 20r1 simulation. This is really not that much of an issue - we need to run over the simulation with a version of the selection with all PID selection removed anyway, so this is fine.

The steps to doing this are:

1. Get a basic job working with the new decay descriptors to book the ntuples we need.
2. Incorporate the GoodPromptFilter class to work with this. Can get this from the Charm production package.
3. Change the stripping line used to be my configured line which mimics the Stripping 21r1 selection without any PID.
4. Run some test jobs on the grid.
5. Run the whole thing on the grid and get it copied to mass storage so the LcTophhBFStuff package can utilise it.

Thus far I've got a local version of DaVinci with the CharmProduction package compiled. The algorithm is recognised by DaVinci. I'm having some serious problems getting the decay descriptor syntaxes working. I'm a bit confused as to why - when running over the data it was easy to update to the new decay descriptors. Now I'm getting bizarre complaints about characters like "[" which are obviously needed in the descriptor.

A Fits to the data

In this appendix I will provide plots of the fits for the subset studies and the full data, along with pull distributions, to indicate that the results are stable.

A.1 Stripping 17b fits to combined polarities

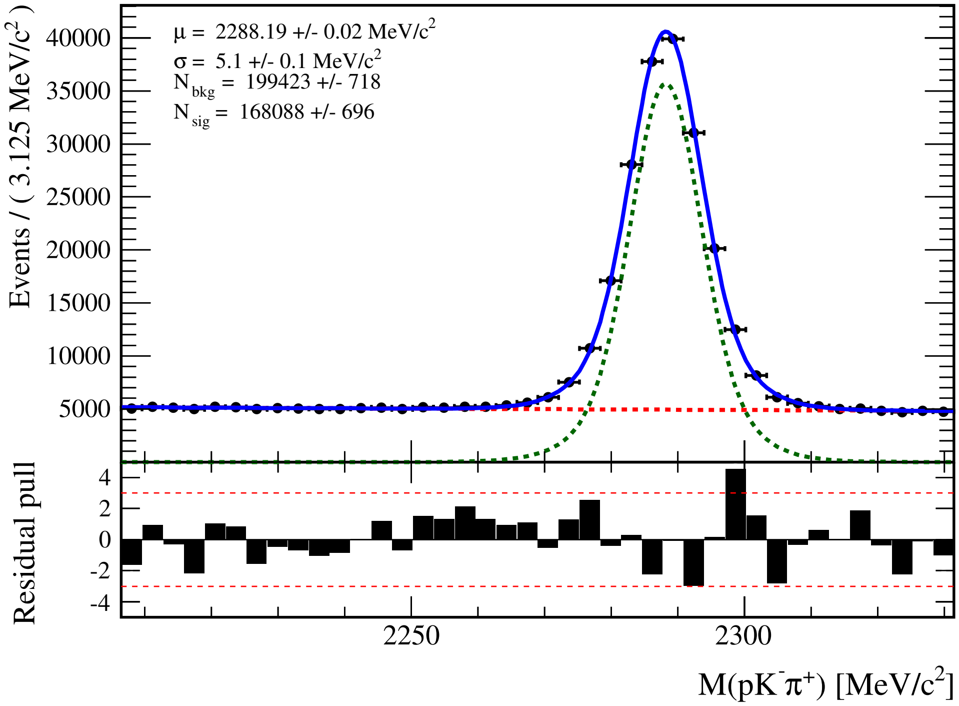

Figure 1: $\Lambda_c^+ \to pK^-\pi^+$- moderate PID - 17b - all

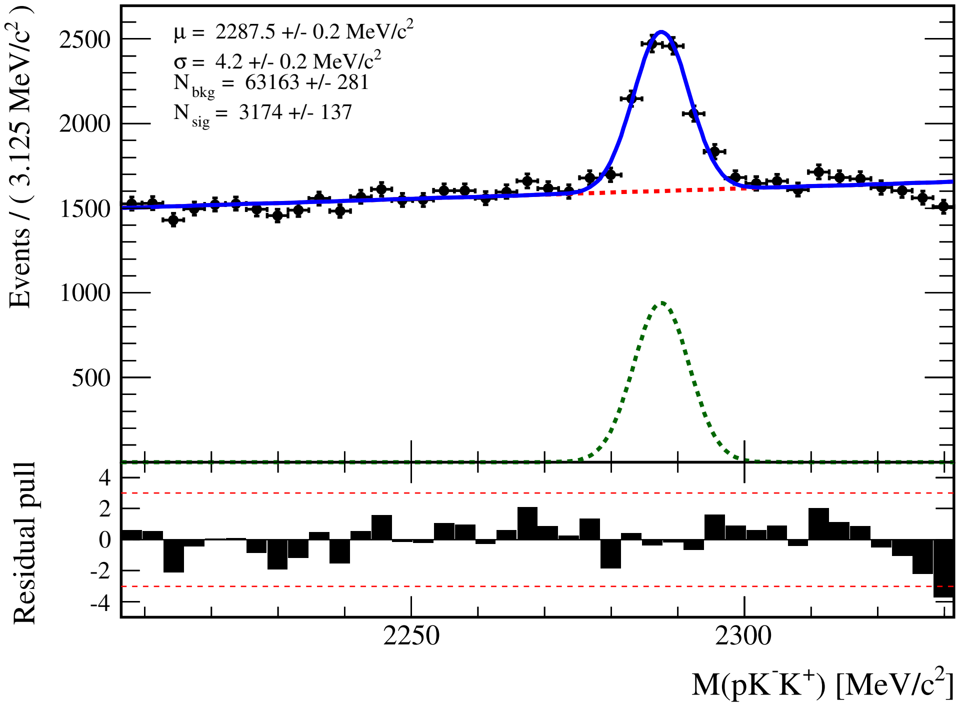

Figure 2: $\Lambda_c^+ \to pK^-K^+$- moderate PID - 17b - all

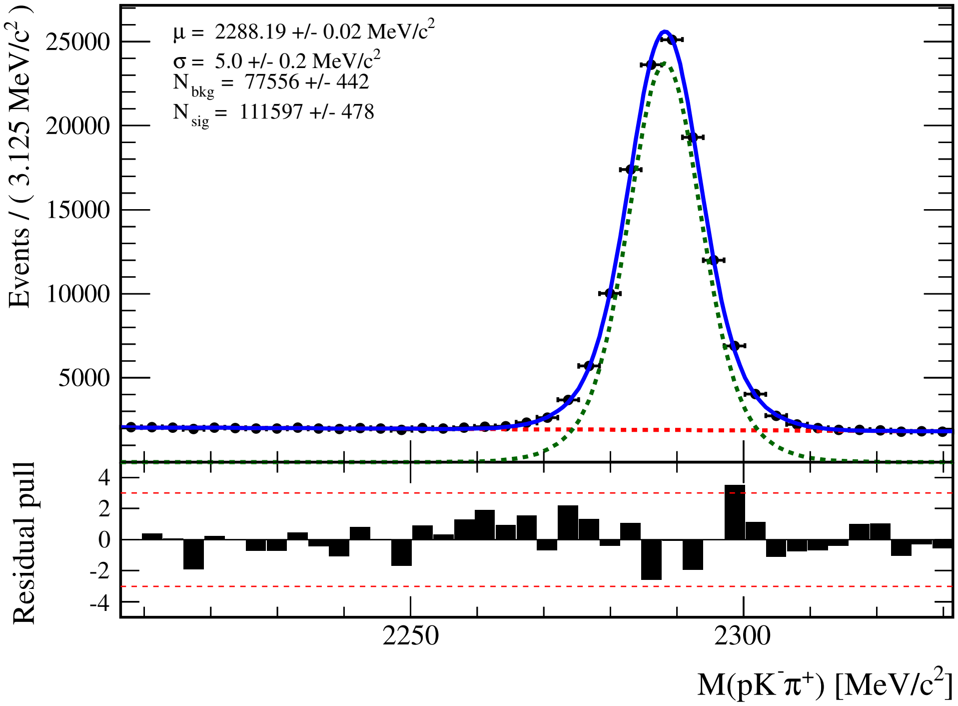

Figure 3: $\Lambda_c^+ \to pK^-\pi^+$- tight PID - 17b - all

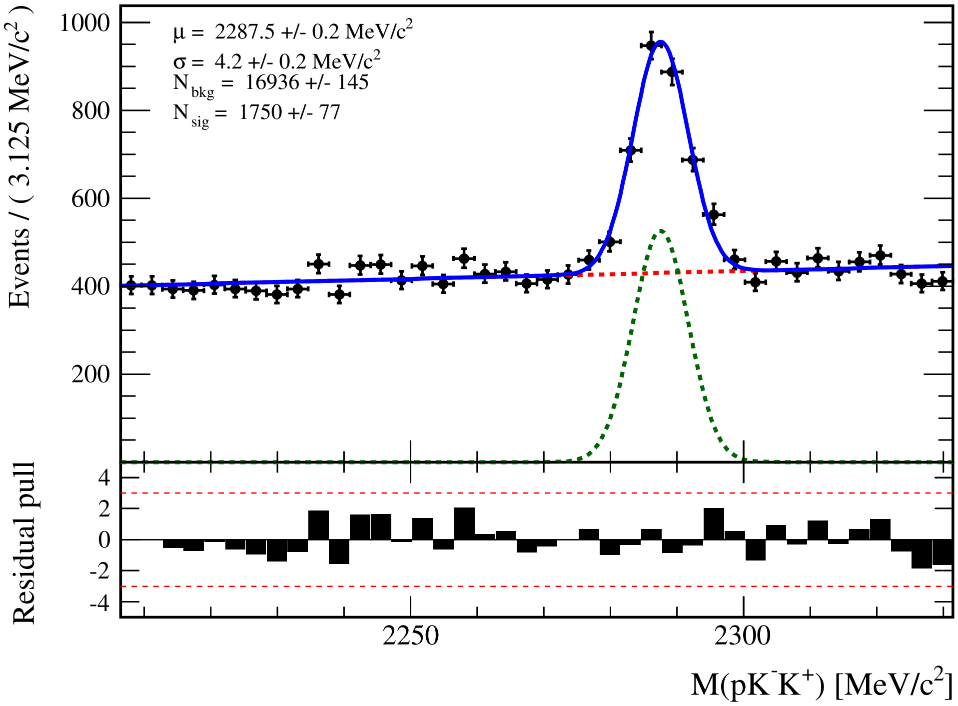

Figure 4: $\Lambda_c^+ \to pK^-K^+$- tight PID - 17b - all

A.2 Stripping 21r1 fits to combined polarities

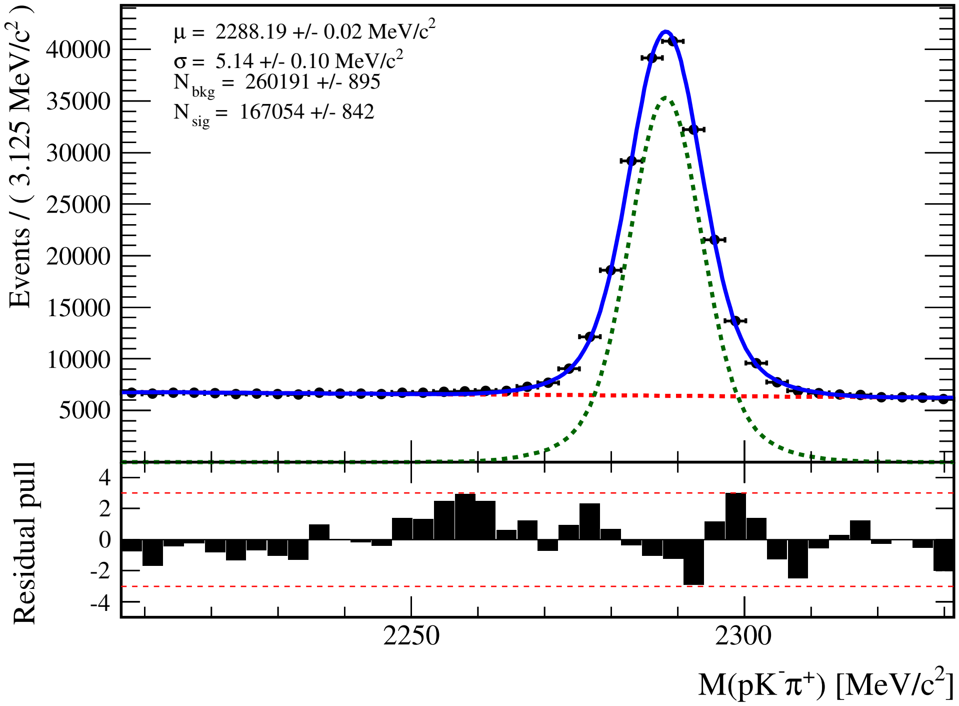

Figure 5: $\Lambda_c^+ \to pK^-\pi^+$- moderate PID - 21r1 - all

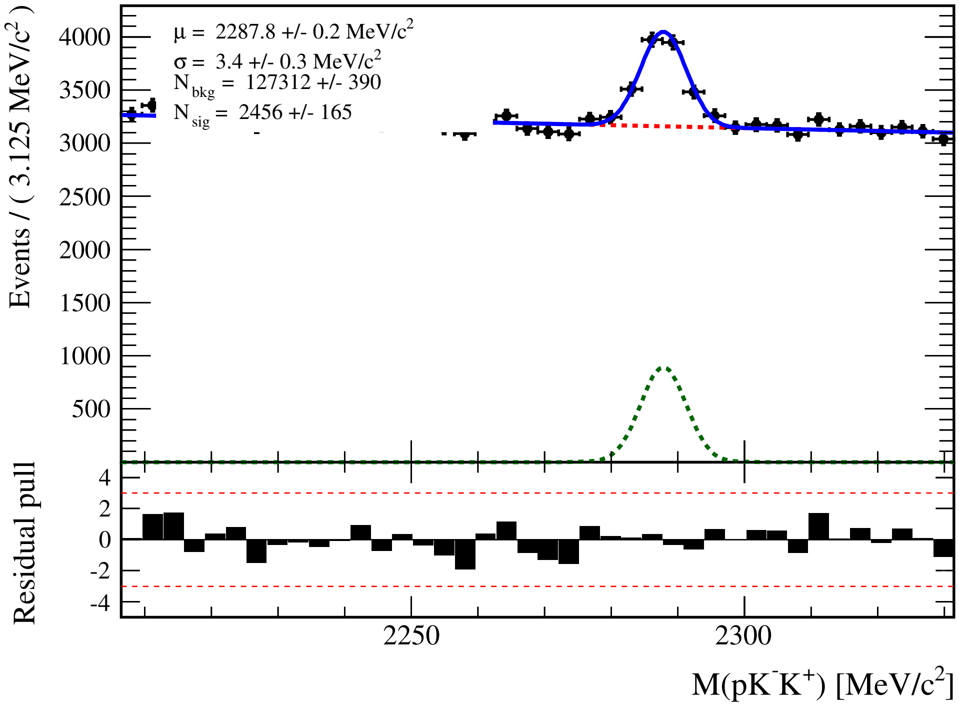

Figure 6: $\Lambda_c^+ \to pK^-K^+$- moderate PID - 21r1 - all

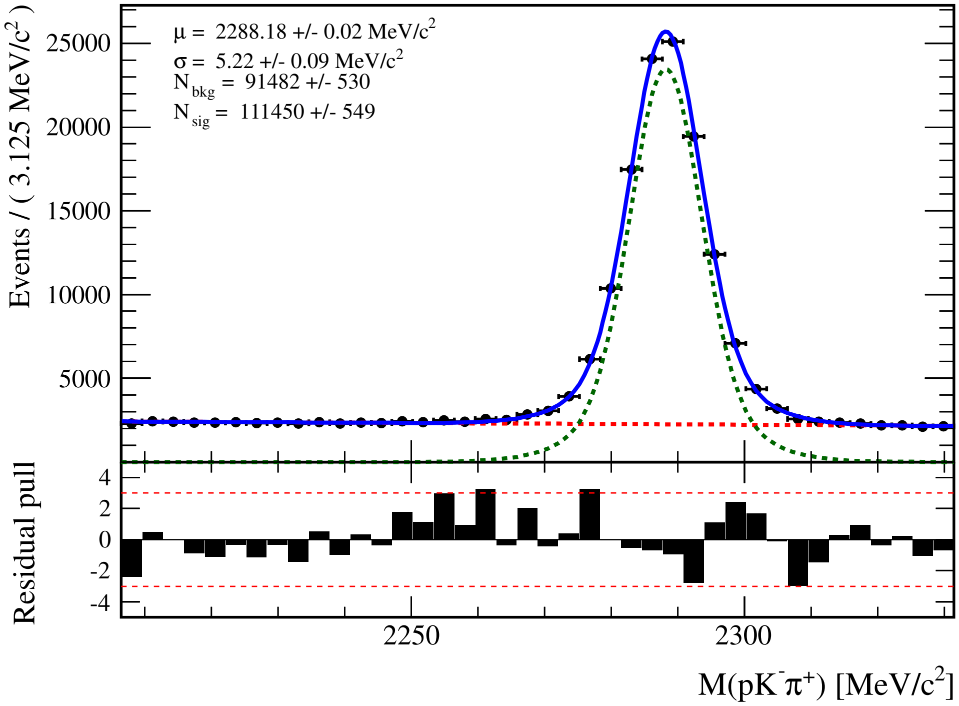

Figure 7: $\Lambda_c^+ \to pK^-\pi^+$- tight PID - 21r1 - all

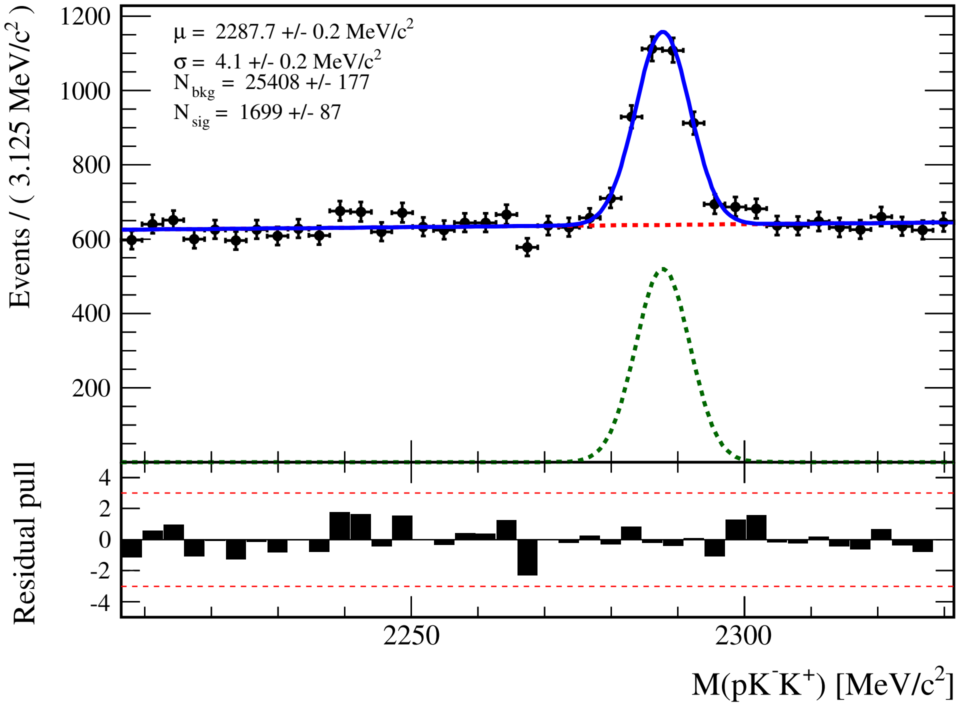

Figure 8: $\Lambda_c^+ \to pK^-K^+$- tight PID - 21r1 - all