

A Search for Doubly Cabibbo Suppressed Decays Using Genetic Programming

Lawrence Berkeley Laboratory

Research Progress Meeting

13 February 2007

Eric W. Vaandering

ewv@fnal.gov

Vanderbilt University and FOCUS Collaboration

Overview

- Introduction to Genetic Programming
 - Populations and Generations
 - Mutation and Crossover
 - Fitness and Natural Selection
- Genetic Programming & Relative BR measurements of doubly Cabibbo suppressed decays
 - $D^+ \rightarrow K^+ \pi^+ \pi^-$ (a check of the method)
 - $\Lambda_c^+ \rightarrow p K^+ \pi^-$
 - $D_s^+ \rightarrow K^+ K^+ \pi^-$
- Systematic uncertainties
 - “Standard” and GP specific
- Conclusions

Motivation

Initial looks at these decays showed not quite enough sensitivity.

Was there a way to improve our search by novel event selection techniques? Can we combine indicators of a good event in a more efficient way?

These analyses are good first test since candidate and normalizing decay modes are so similar.

GP also looked like a method that might lend insight into our data.

Keep an open mind about new ideas and how we might use them.

GP Fundamentals

To solve problems, maybe we should emulate biology and the evolutionary process. (→ Genetic Algorithms)

Since we will use computer programs to implement our solutions, maybe the *form* of our solution should *be* a computer program. (→ Genetic Programming)

- Applies a model of biological evolution to program “discovery”
- Iterative procedure on populations of programs
- Pioneered by John Koza, seminal reference: *Genetic Programming: On the Programming of Computers by Natural Selection* (1992)
- Since 1992, more than 4,000 papers applied to a wide range of problems

Populations and Generations

This discussion may be a little detailed, but keep a few things in mind:

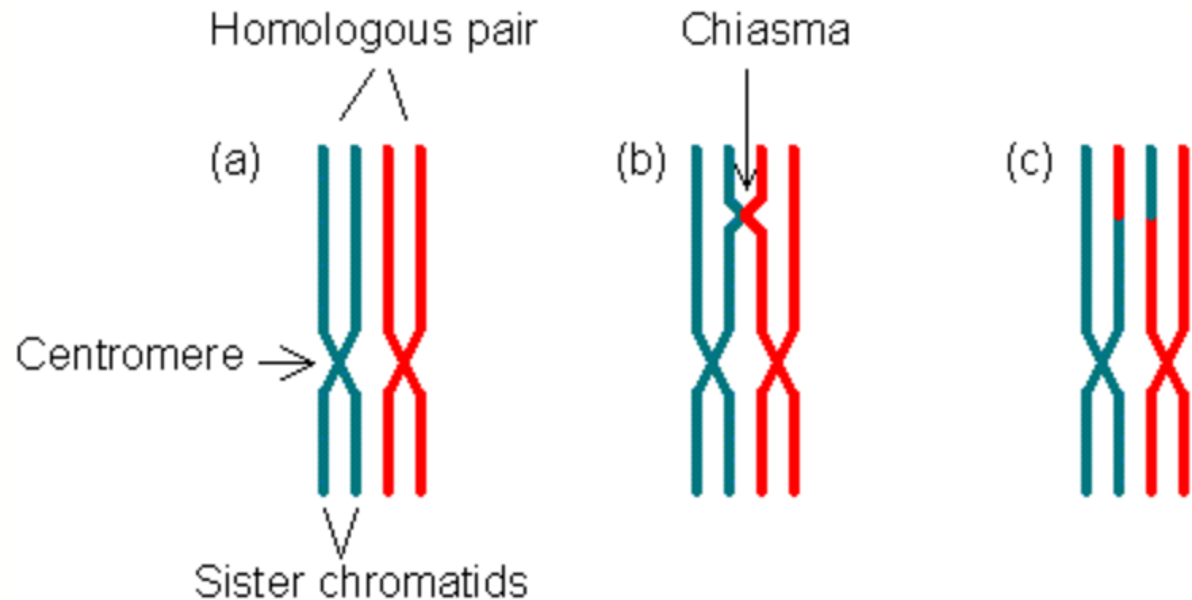
- Program \equiv Organism
- Program will filter events
- We need a way to know if one filter is better than another

Genetic Programming works by transforming one group of programs (filters) in generation n into another group of programs in generation $n + 1$. There are typically a few hundred to a few thousand programs per generation.

Gene Cross-over and Mutation

There are two processes which, combined with natural selection, drive biological evolution.

1) Biological (DNA) Cross-over



2) Mutations in nature change the genetic code for a small region of DNA. Usually are harmful or neutral; occasionally helpful (better/different organism arises).

Preparatory Steps

To prepare to solve a problem with Genetic Programming, two steps are necessary:

- Define a group of functions (\equiv genes)
 - Some functions may provide an input
 - Other functions may perform an operation
 - $+$, $-$, $>$, $<$ are all “functions”
 - So are IF-THEN-ELSE
 - Output of one function is input to another
- Determine a way to separate good programs from bad

How I'll use this:

- Input values: N-tuple variables that describe an event
- For each event: final output value \rightarrow keep or discard event
- How “good” is the collection of kept events?

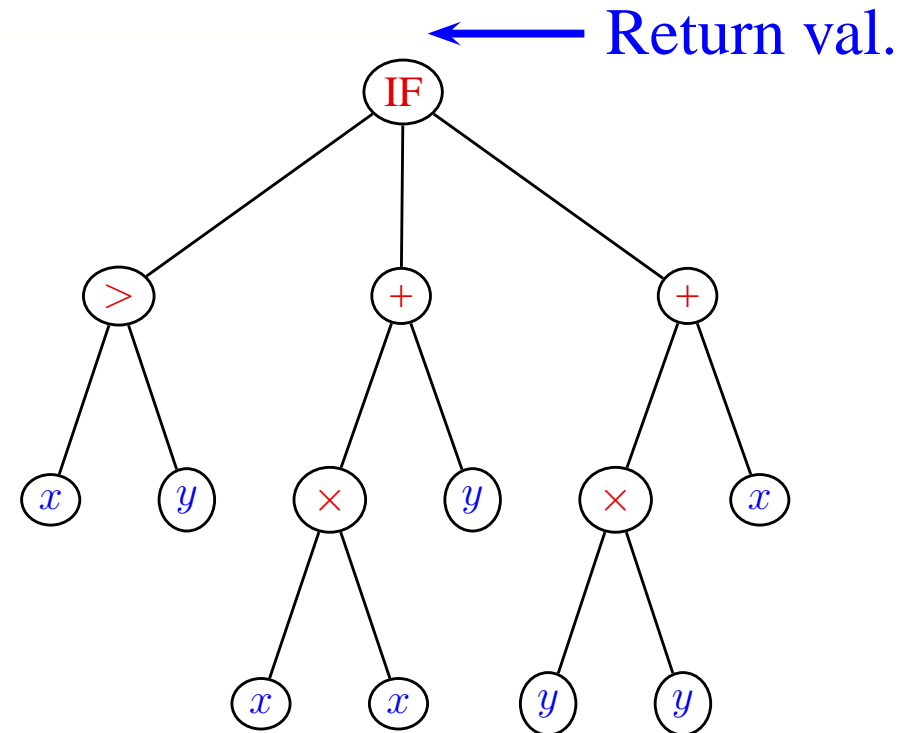
Tree Representation

Genetic Programming fundamentals are easier to illustrate if we adopt a “Tree” representation of a program. An example of this representation:

Code:

```
float myfunc(float x, float y) {  
    float val;  
    if (x > y) {  
        val = x*x + y;  
    } else {  
        val = y*y + x;  
    }  
    return val;  
}
```

Program tree

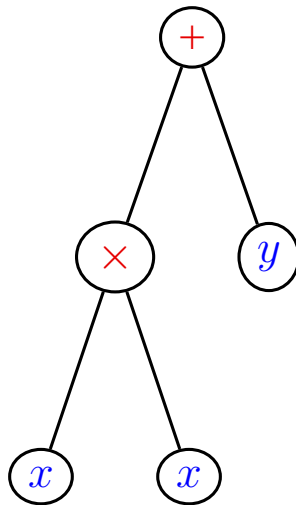


Tree Representation, cont.

From a fraction of our tree, we can see a few things:

Two kinds of “nodes”

- There are functions (IF, $>$, $+$, $*$)
- There are “terminals” (x , y)



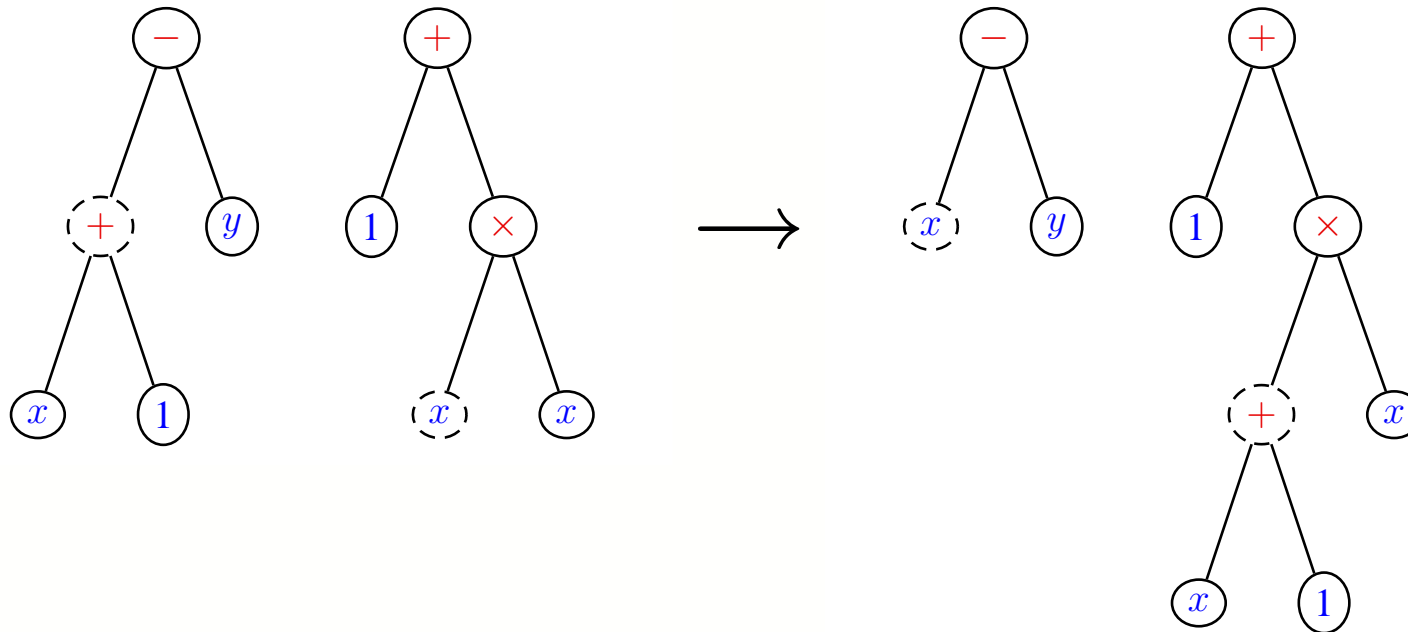
If we allow *any* function or terminal at any position, then all operations must be defined:

- IF (float)
- $x + (y > x)$
- Divide by zero (if we use division)

A population of trees is built randomly choosing functions or terminals until all branches are terminated.

Crossover (Recombination)

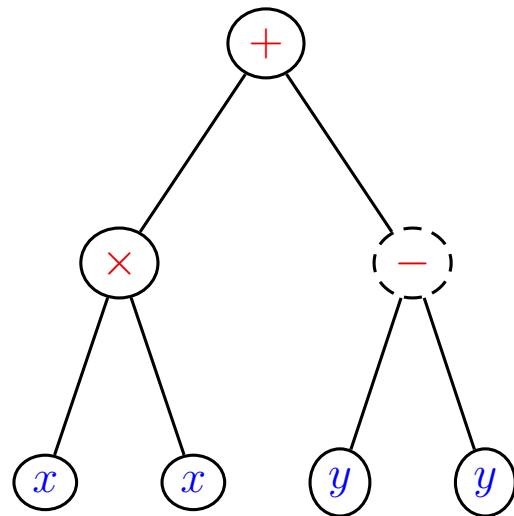
Two programs and crossover points within them are chosen. Sub-trees are removed and swapped between trees, giving two new “children”



It may combine the best aspects of both parents into one child (of course, we are just as likely to end up with the worst aspects in one child).

Mutation

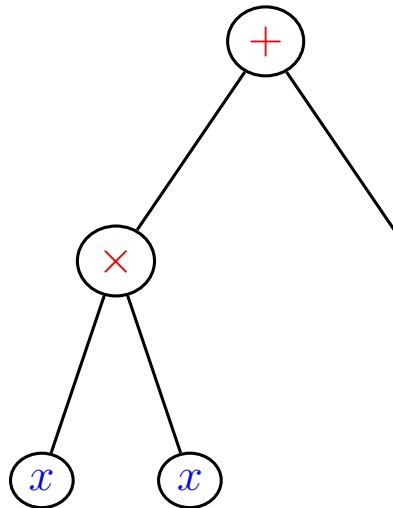
Occasionally we want to introduce a mutation into a program or tree.



Pick a parent & mutation point

Mutation

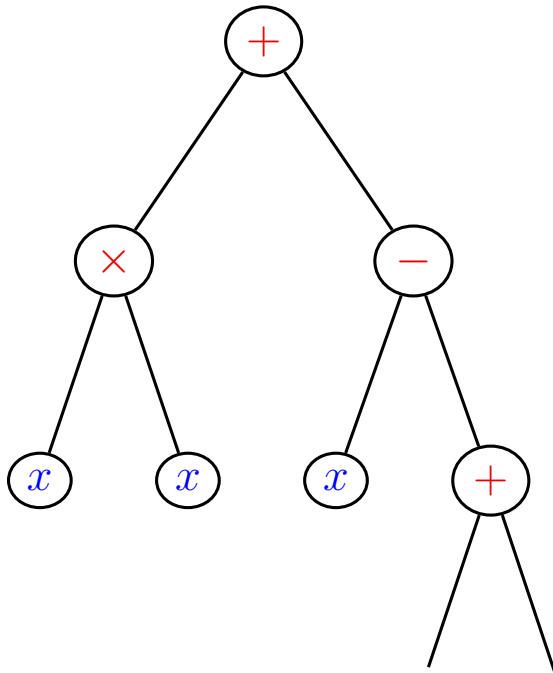
Occasionally we want to introduce a mutation into a program or tree.



Pick a parent & mutation point
Remove the subtree

Mutation

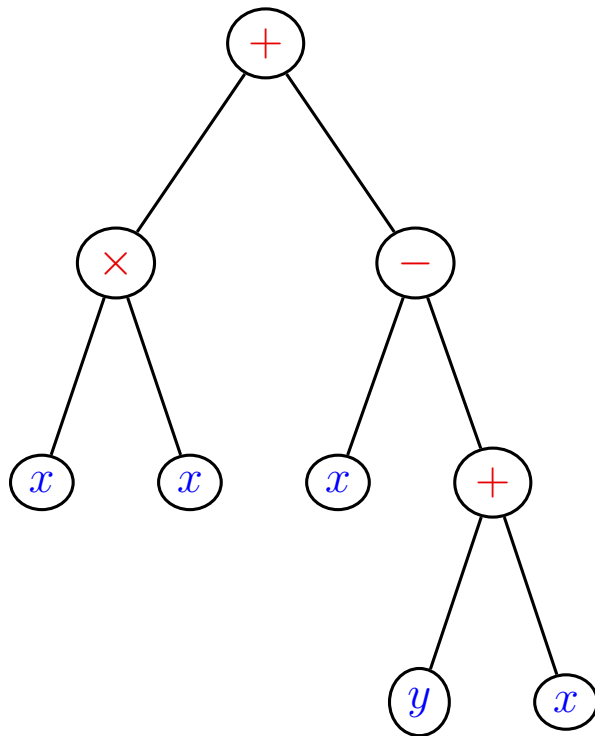
Occasionally we want to introduce a mutation into a program or tree.



Pick a parent & mutation point
Remove the subtree
Build a new subtree as if it were
a “root” tree

Mutation

Occasionally we want to introduce a mutation into a program or tree.



Pick a parent & mutation point
Remove the subtree
Build a new subtree as if it were
a “root” tree

Mutation can often be very de-
structive in Genetic Program-
ming

Remember: Input variables describe event. Program/tree output variable classifies event.

Practical considerations

Obviously, a tree can grow nearly infinite in size. This is usually undesirable. There are ways to control this:

- Set limits on number of nodes
- Set limits on depth of nodes
- Initial, randomly created trees are kept small

So far we've mimicked *how* organisms reproduce.
The other half of the problem is *which ones* reproduce.

Survival of the Fittest

In nature, we know that the more fit an organism is for its environment, the more likely it is to reproduce. This is one of the basic tenets of evolutionary theory.

- Organisms with serious deformities are still-born or die at a young age
- Faster, stronger, or longer lived organisms will produce more offspring than “normal” organisms



We see this behavior in nature all the time

Survival of the Fittest in GP

The Genetic Programming method mimics survival of the fittest by determining a numerical *fitness* for each program. Which programs reproduce is based on this fitness.

Possibilities:

- How many events does it classify correctly?
- In how many cases does it provide the correct output?
- How well does it fit the data?

Considerations:

- The problem *must* allow for inexact solutions. There may be a single *correct* program, but there must be a way to distinguish between increasingly incorrect programs. (Otherwise we are engaging in a random search.)
- Pathological programs → very poor fitness (still-born)

Reproduction Probabilities

Programs are chosen randomly to “breed” and populate the next generation according to their fitness. We use something like a roulette wheel where the size of the slot is proportional to the fitness of the program.



Crossover: Spin wheel twice to select two parents.

Mutation: Spin wheel once to select one parent.

Some parents will be selected multiple times, some not at all.

- The best program is *most likely* to be chosen, but is *not guaranteed* to be chosen
- The worst program *may* be chosen

Running the GP

Putting this all together, we are ready to “run” the GP (find a solution).

- User has defined functions, variables, and measure of fitness
- Generate a population of programs (few hundred to few thousand)
- Test each program against fitness definition
- Choose genetic operation (crossover/mutation) and parent programs to create next generation, randomly according to fitness
- Repeat this process generation after generation
 - Often tens of generations are needed to find the best program
- At the end, we have a large number of programs; look at the best one found

Application to HEP

Ok, so all this may be interesting to computer scientists, but how does it apply to High Energy Physics?

In FOCUS, we typically select interesting (signal, we hope) events from background processes using cuts on interesting variables. That is, we construct variables *we* think are interesting, and then require that an event pass the AND of a set of selection criteria.

Instead, what if we give a Genetic Programming framework the variables we think are interesting, and allow *it* to construct a filter for the events?

- If an AND of cuts is the best solution, the GP can find that

This is not such a radical approach. *E.g.*, neural networks and boosted decision trees are used effectively in many experiments.










Questions

When considering an approach like this, some questions naturally arise:

- Does it do as well as traditional methods do?
 - Can we find new cuts for traditional analyses? Yes!
- How do we know it's not biased?
 - We do optimize on data (but not on the signal we are interested in)
 - To check, all optimizations are on half the events
- The tree can grow large with useless information.

Flavor Physics in 2 Minutes

Quarks (and hadrons) *decay* weakly by emitting a virtual W boson which couples the up and down type quarks. Because weak and mass eigenstates are mixed, decays cross generations. We use the CKM matrix to describe these mixing couplings.

d	s	b	
			u
			c
			t

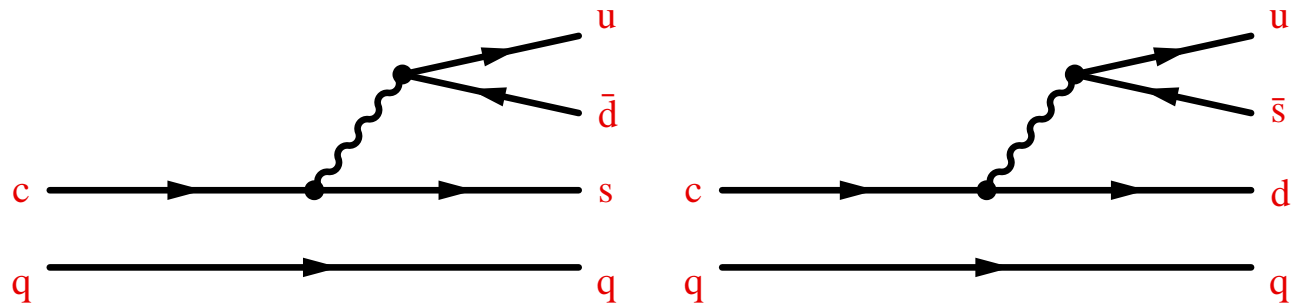
We can see that the charm (c) quark is coupled mostly to the strange (s), a little to the down (d), and very little to the bottom (b) quarks.

Of course, $c \rightarrow b$ transitions are not kinematically allowed.

Historically, the 2×2 sub-matrix was parameterized by the Cabibbo angle, θ_c with elements of $\sin \theta_c$ and $\cos \theta_c$. Since the decay coupling is squared, the probability of a $c \rightarrow d$ transition relative to $c \rightarrow s$ is $\sin^2 \theta_c / \cos^2 \theta_c = \tan^2 \theta_c \approx 5\%$.

Cabibbo Suppressed Decays

Doubly Cabibbo suppressed decays can only be observed in charm. Both W vertices are Cabibbo suppressed.



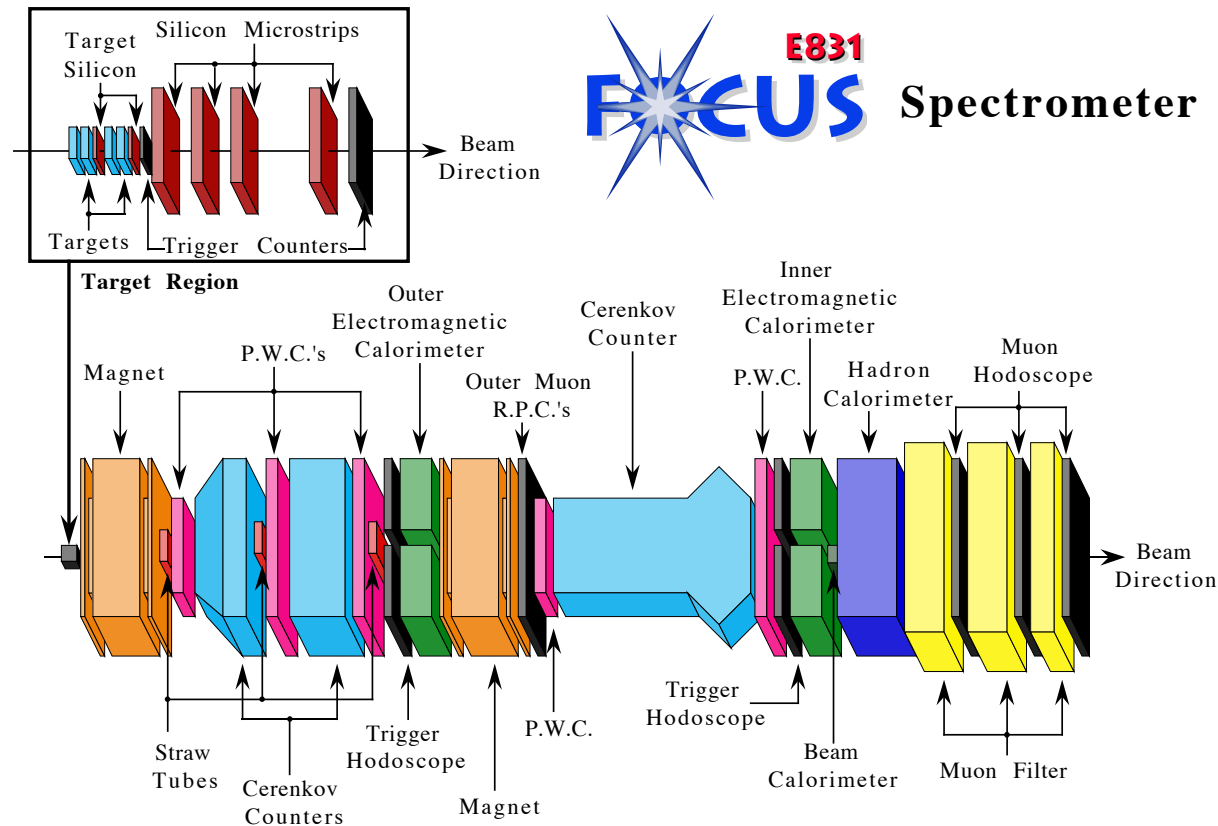
Cabibbo Favored

Doubly Cabibbo Suppressed

Doubly Cabibbo suppressed decays are chosen as a first test of the method since the final state particles are often similar (*e.g.*, $\Lambda_c^+ \rightarrow pK^- \pi^+$ vs. $\Lambda_c^+ \rightarrow pK^+ \pi^-$). This eliminates many possible sources of systematics arising from inexact modeling of the selection process.

Expected relative branching ratios: $\sim \tan^4 \theta_c \approx 0.25\%$.

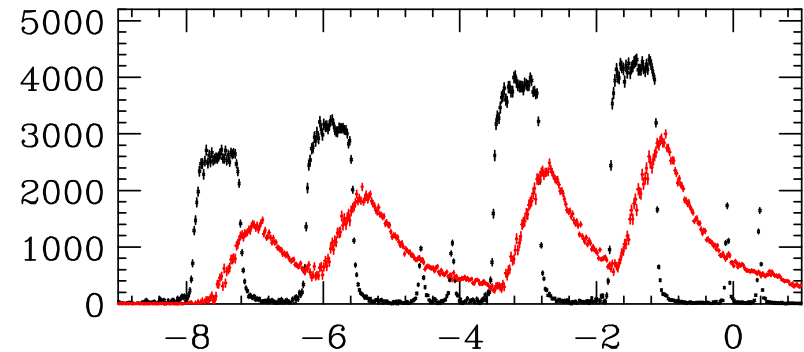
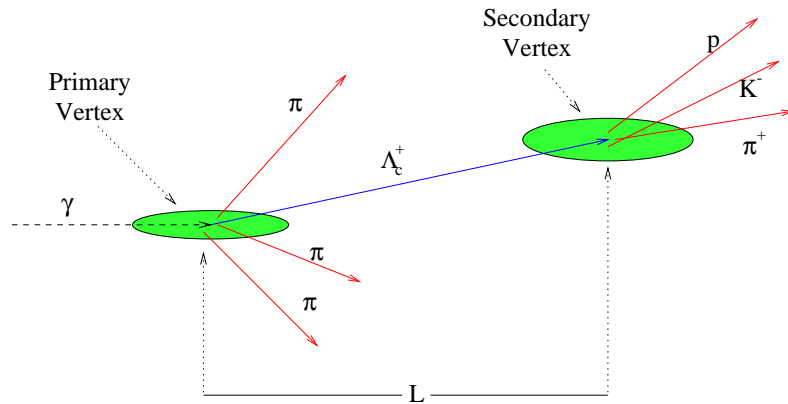
FOCUS Spectrometer



Highlights:

- γ beam on segmented target
- Silicon vertexing
- EM/hadronic calorimeters
- MWPC tracking
- Threshold Čerenkov
- Muon detectors

Target and Vertexing

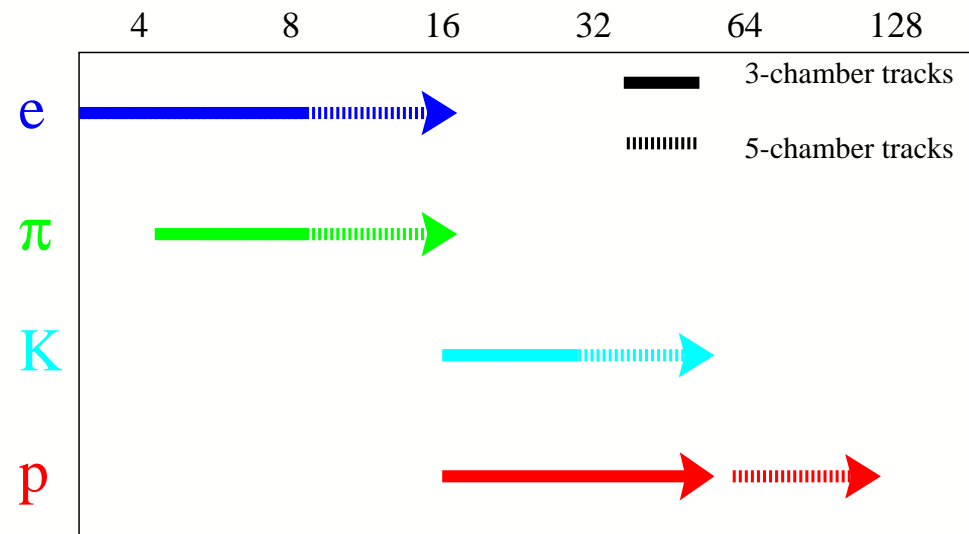


Some details of the FOCUS candidate driven vertexing

- ℓ : Distance between production and decay vertices. ℓ/σ_ℓ , significance of separation (signature for weak decays)
- OoT: Significance of decay being out of target material
- CLP, CLS: Conf. Levels of production and decay vertices
- Iso1: CL that tracks from decay vertex are consistent with production vertex
- Iso2: CL that other tracks (incl. from production vertex) are consistent with decay vertex

Čerenkov System

- Three multi-cell threshold Čerenkov detectors
- Gives positive particle ID over limited momentum ranges



Positive ID ranges for tracks (GeV/c)

- We use an ID method based on log-likelihoods
 - For each particle hypothesis, find probability of observing the # of photons seen
 - Calculate differences between particle hypotheses
 - Gives a continuum of particle ID values

Variables and Operators

Give the GP lots of things to work with:

Functions (22)		Variables ($D^+ - 38, D_s^+ - 37, \Lambda_c^+ - 40$)		
\times	sign	ℓ	$\Delta W(\pi p)$	0,1
/	negate	σ_ℓ	$\Delta W(Kp)$	#dau
+	max	ℓ/σ_ℓ	$\Delta W(\pi K)$	σ_t
-	min	OoT	π_{con}	p_T
x^y	NOT	CLS	$\Delta W(K\pi)$	Σp_T^2
$\sqrt{\quad}$	AND	CLP	Track χ^2 's	m_{err}
log	OR	Iso1	OS Vertex CL	μ_{max}
>	XOR	Iso2	OS $\Delta W(\pi K)$	TS/NoTS
<	IF	#life	OS CL_μ	REME
$\langle \Rightarrow \rangle$	sin	Pri. OoT	Real (-2, +2)	DLink
$f(n)$	cos	$p(\Lambda_c^+ / D_s^+ / D^+)$	Int (-10, +10)	$\Sigma_c^{(*)}$ tags

One typical program: 80 nodes (40 func., 40 var.)

$$\rightarrow 40^{22} \times 40^{40} = 2 \cdot 10^{99} \text{ combinations.}$$

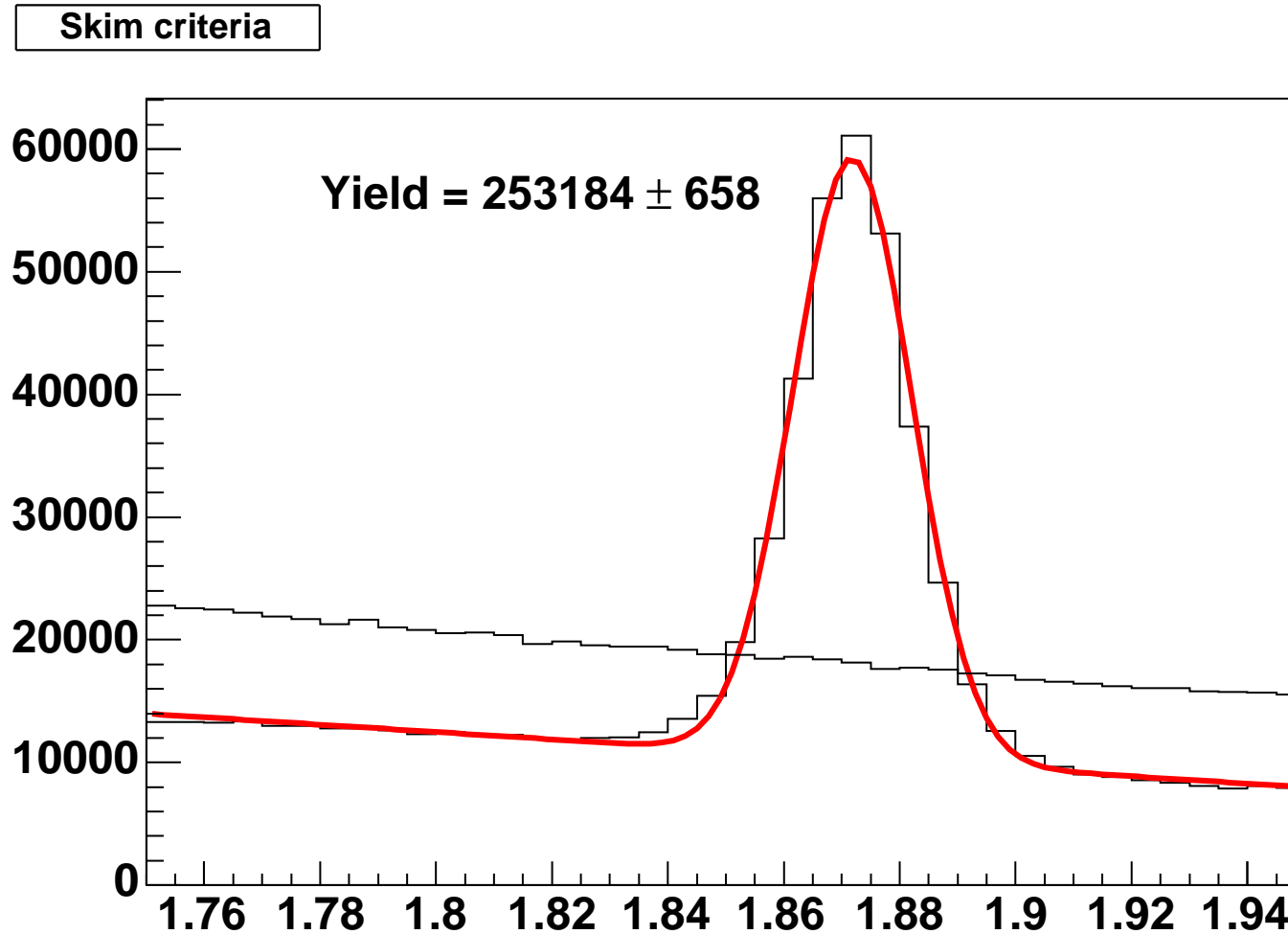
$$D^+ \rightarrow K^+ \pi^+ \pi^-$$

While $\Lambda_c^+ \rightarrow p K^+ \pi^-$ and $D_s^+ \rightarrow K^+ K^+ \pi^-$ are the decays we're pursuing with this method, $D^+ \rightarrow K^+ \pi^+ \pi^-$ is a useful check.

This branching ratio has been measured and is surprisingly large (almost $3 \tan^4 \theta_c$). The PDG value is $0.68 \pm 0.08\%$ relative to $D^+ \rightarrow K^- \pi^+ \pi^+$.

A FOCUS Dalitz and branching ratio measurement dominates this ($0.65 \pm 0.08 \pm 0.04\%$), so let's see how Genetic Programming fares vs. traditional methods

Signals Before Optimization



Fit shows $D^+ \rightarrow K^- \pi^+ \pi^+$ normalizing mode
“Linear” histogram is DCS candidates

Evaluating the GP

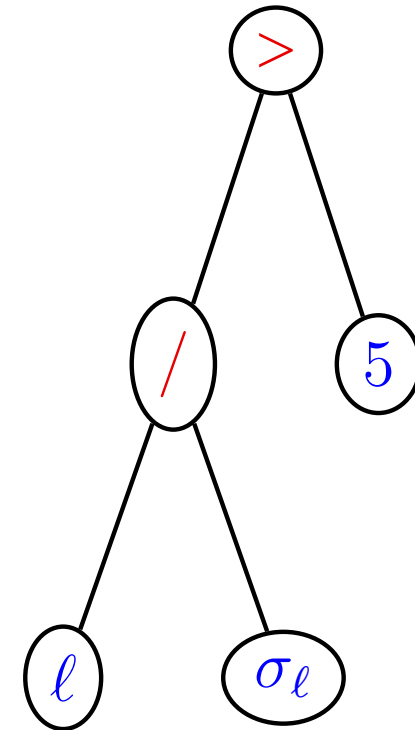
We use GP generated filters to classify each event as “pass” or “fail.” We use the same filter for both CF and DCS decays and only consider “pass” events.

Pass \equiv return value > 0 .

Want to predict significance of DCS signal:

$$\frac{S_{\text{DCS}}}{\sqrt{S_{\text{DCS}} + B_{\text{DCS}}}} \propto \frac{S_{\text{CF}}}{\sqrt{B_{\text{DCS}}}}$$

$\nearrow c \cdot S_{\text{CF}}$
 $\searrow \approx 0$



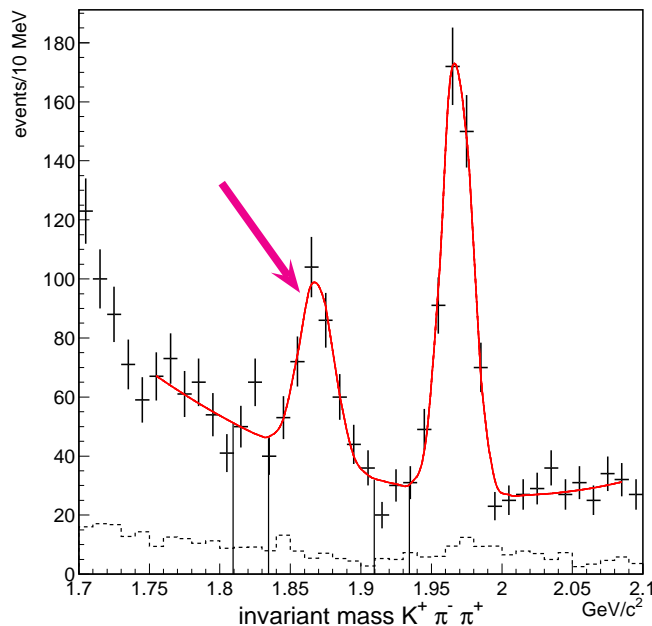
A very simple tree, just a cut on $l/\sigma_l > 5$

Assume effect of filter on CF and DCS signal events is the same

Find S_{CF} and B_{DCS} from fits to the data ($\frac{1}{2}$ events, DCS signal blinded)

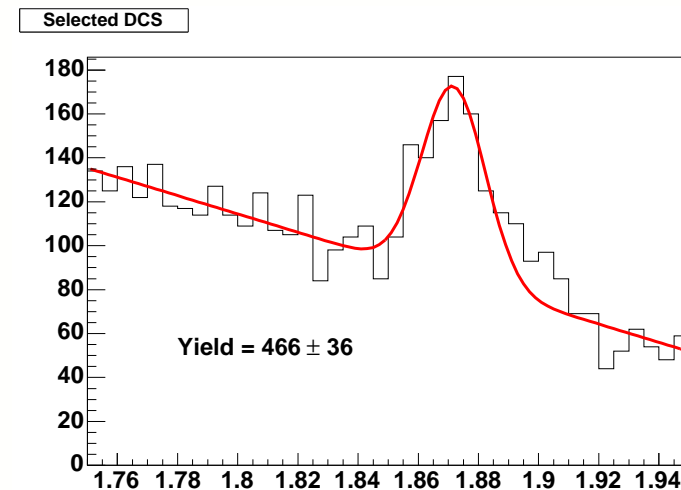
Comparison with Cut Analysis

- From PLB 601:10–19, measured BR of $D^+ \rightarrow K^+ \pi^+ \pi^-$
 - Rel. BR — PLB: $0.65 \pm 0.08 \pm 0.04$, GP: 0.76 ± 0.06
- Not a perfect comparison, not optimized on $S/\sqrt{S+B}$
- Extra corrections and studies in published analysis

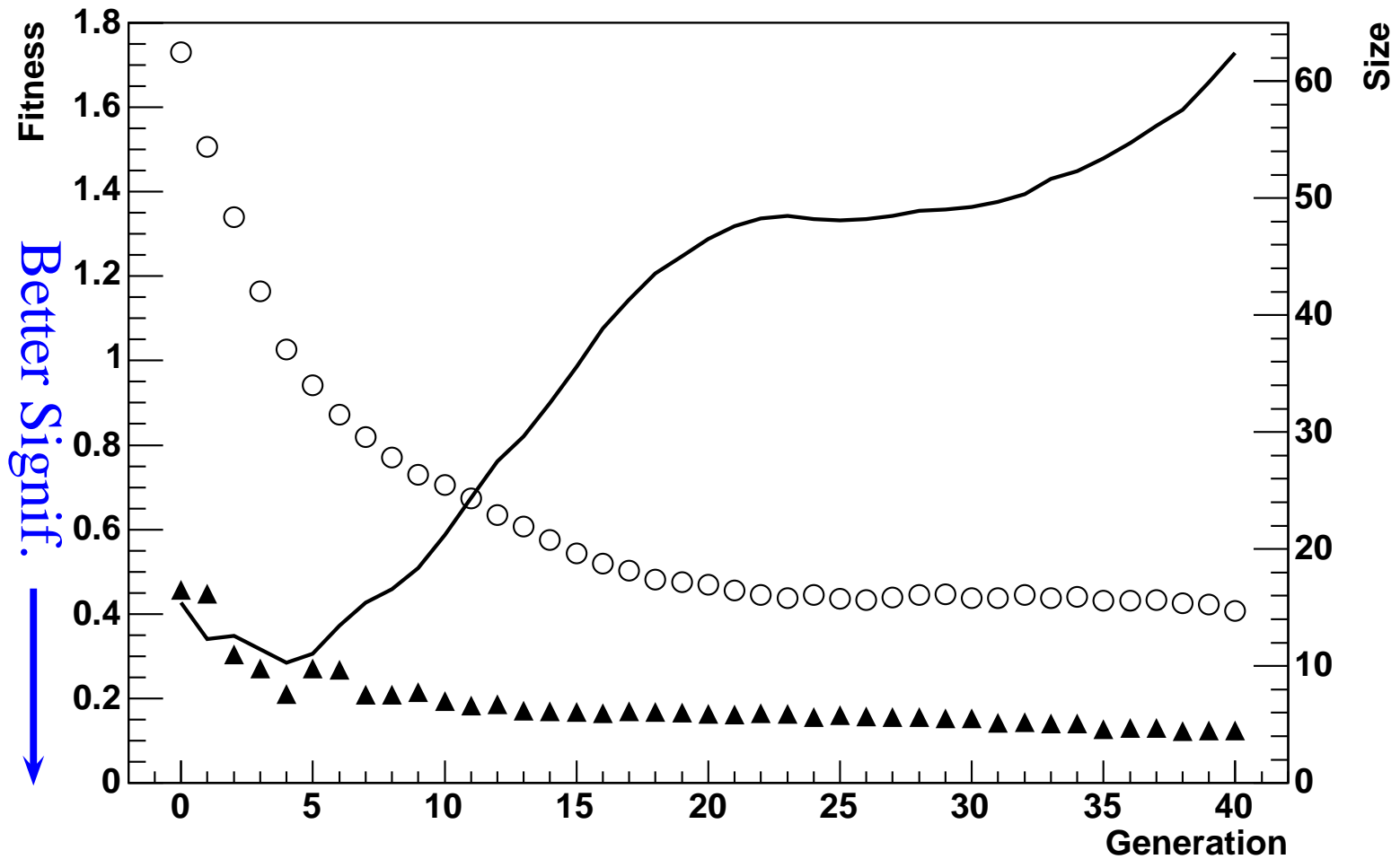


SCS D_s^+ decay on right

- Similar signal to noise
- Cuts: Yield = 189 ± 24 events
- GP: Yield = 466 ± 36 events



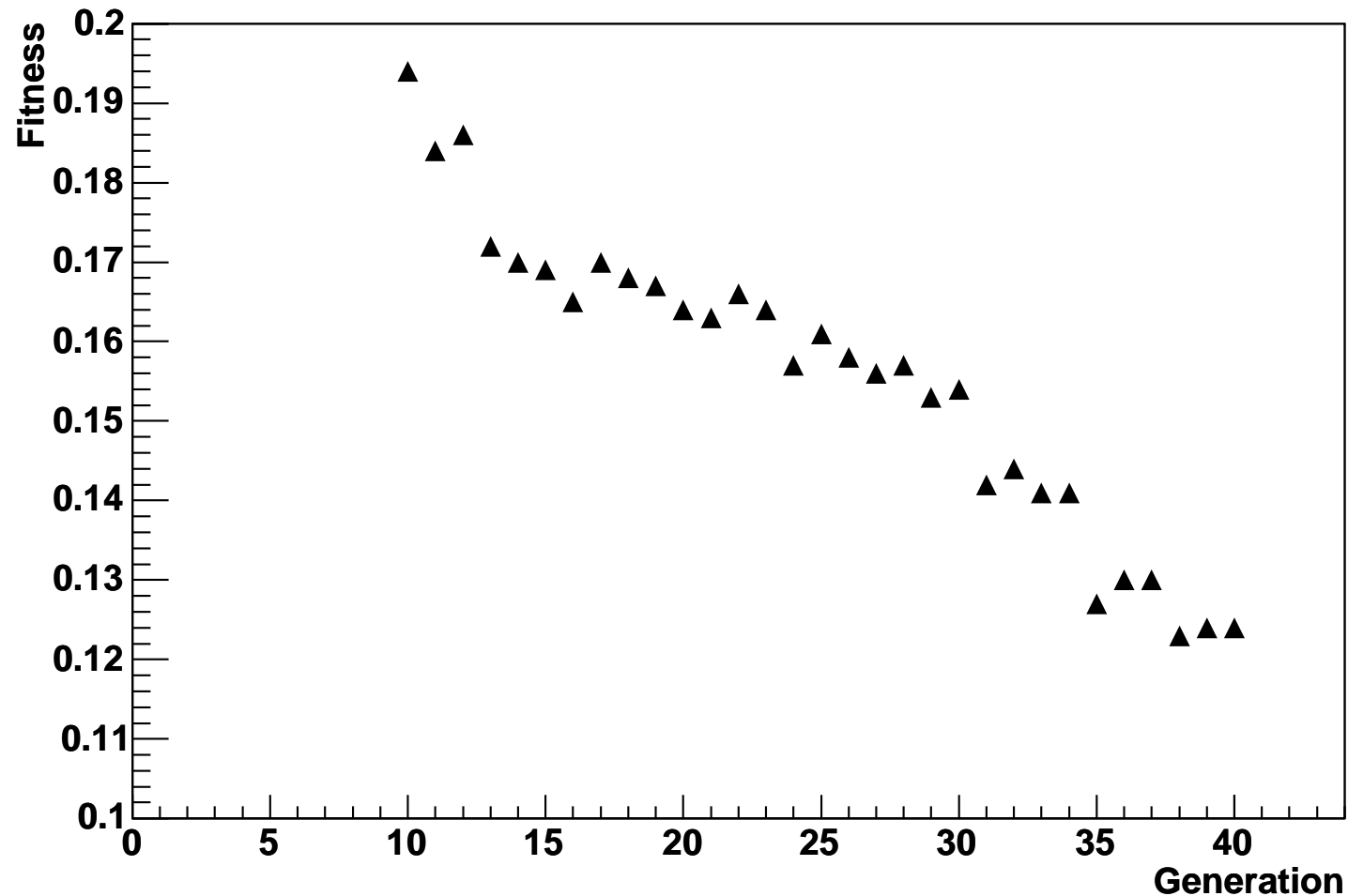
Evolution of Population



Circles: Avg. fitness
Diamonds: Best Fitness

Solid line: Avg. Size

Evolution of Best Programs

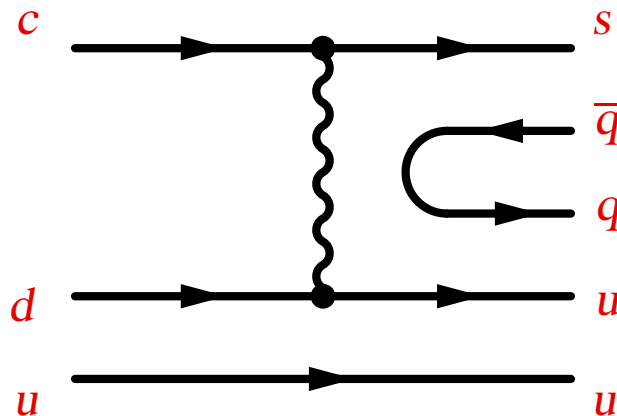


Still evolving at 40 generations

$$\Lambda_c^+ \rightarrow p K^+ \pi^-$$

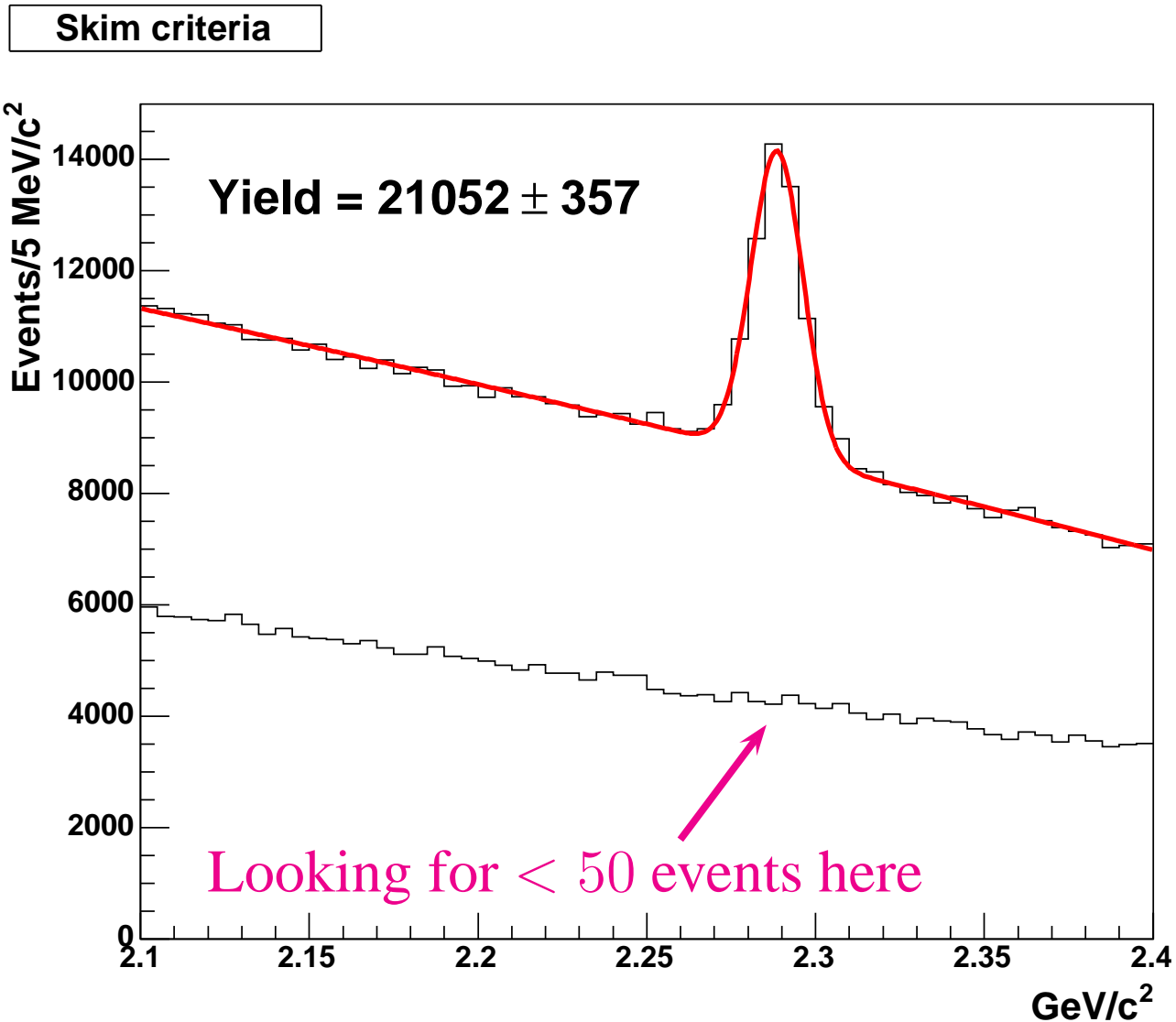
The first decay we search for is $\Lambda_c^+ \rightarrow p K^+ \pi^-$. There are no observations or limits. Even an observation of $\tan^4 \theta_c$ relative to $\Lambda_c^+ \rightarrow p K^- \pi^+$ is challenging for FOCUS, but it's even more difficult.

The Cabibbo favored mode has an W -exchange contribution while the DCS decay does not. (This contribution causes the Λ_c^+ lifetime to be about half the Ξ_c^+ (csu) lifetime.)



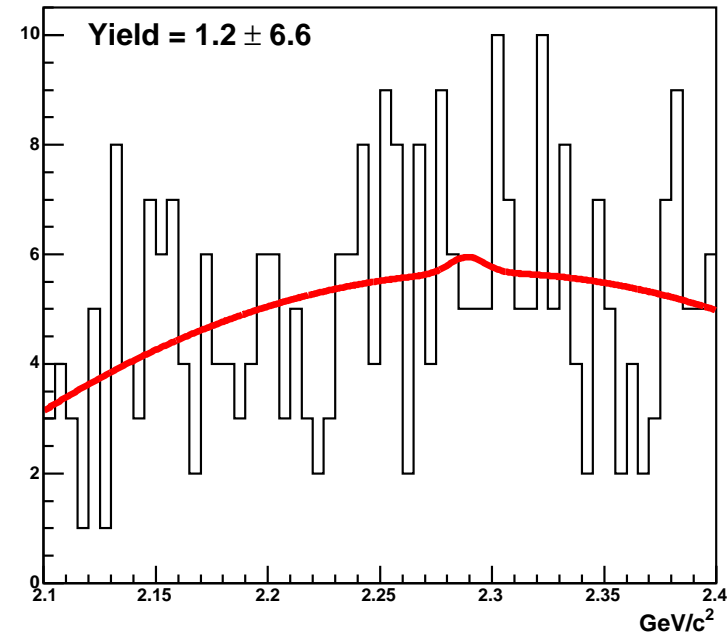
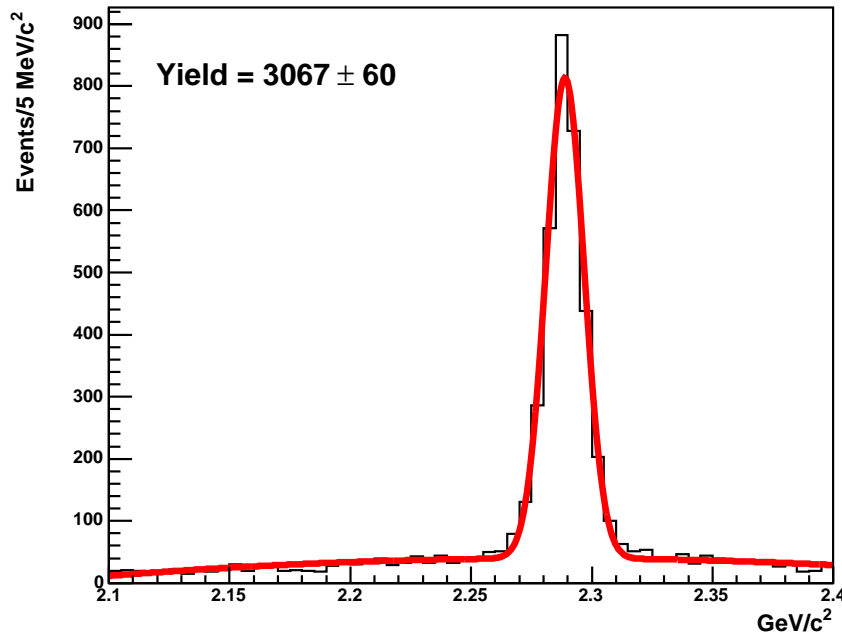
This means the expected branching ratio will be reduced.

Pre-GP selection Λ_c^+ signals



Lower histogram: $\Lambda_c^+ \rightarrow pK^+\pi^-$ candidates

Signals after GP selection



GP retains 3,000 of 21,000 original CF events
DCS background reduced $1000\times$

Fits during optimization use 1st degree polynomial.

Comparing with Cut Method

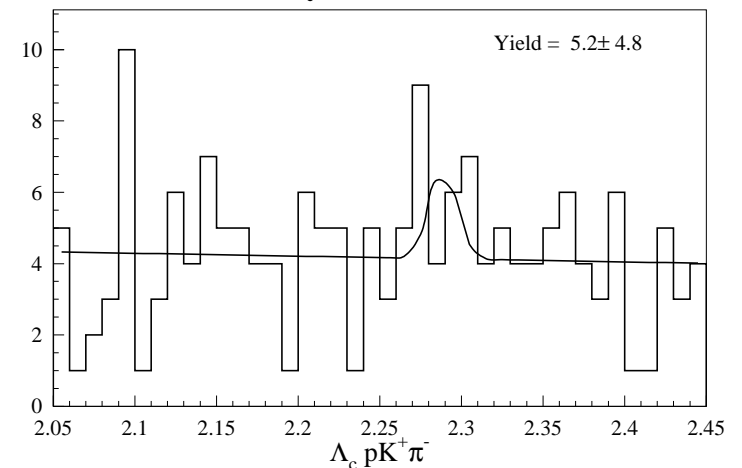
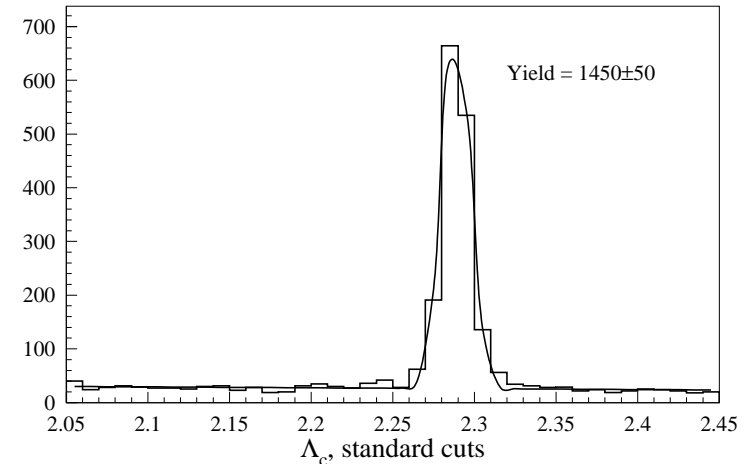
Compare this analysis with an old attempt with normal method:

GP found 3070 ± 60 CF and
 1.2 ± 6.6 DCS events

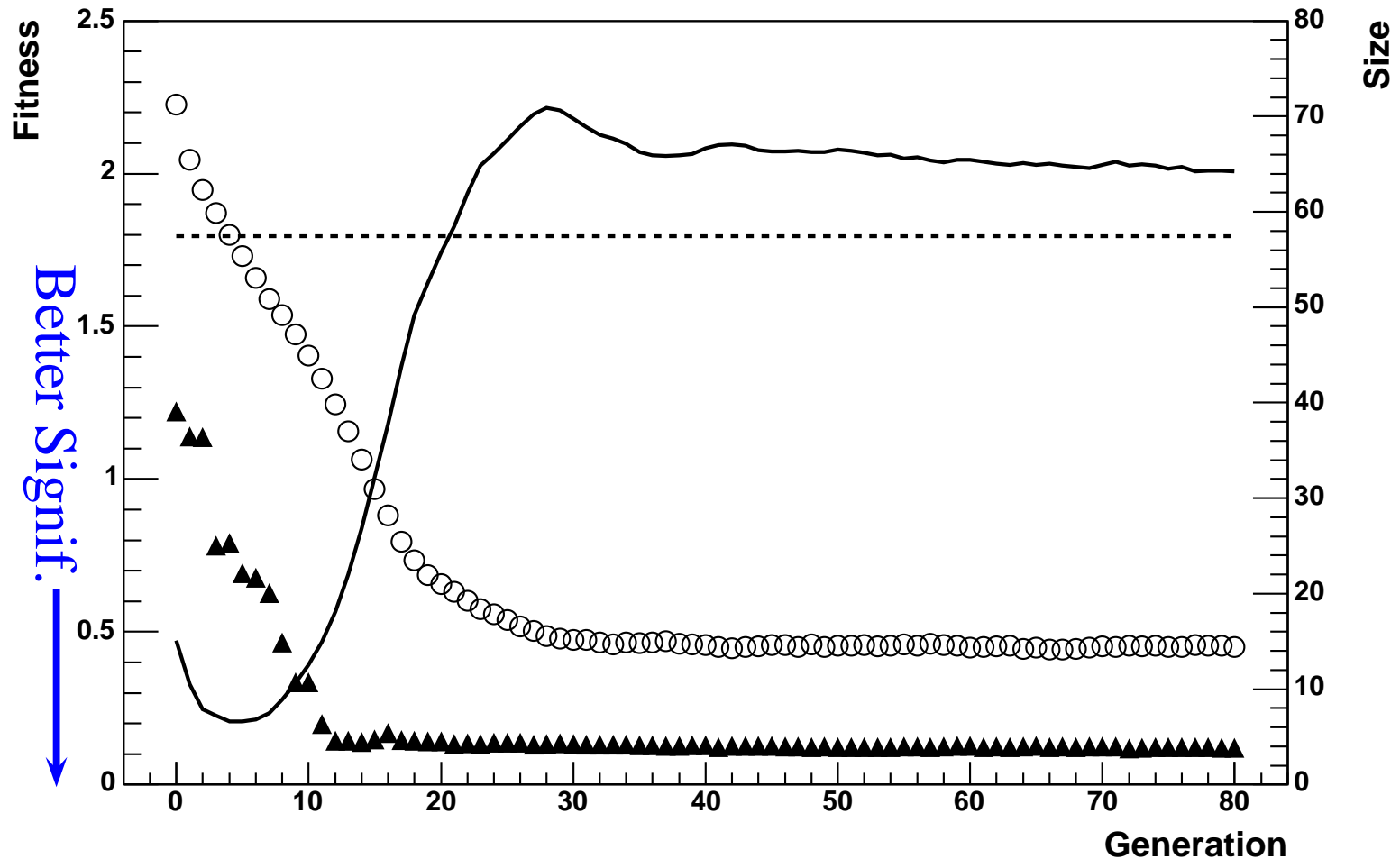
Figure of merit is $\sigma Y_{\text{DCS}}/Y_{\text{CF}}$

$$\frac{6.6}{3070} \text{ vs. } \frac{4.8}{1450}$$

is 0.21% vs. 0.33%
(a significant improvement)



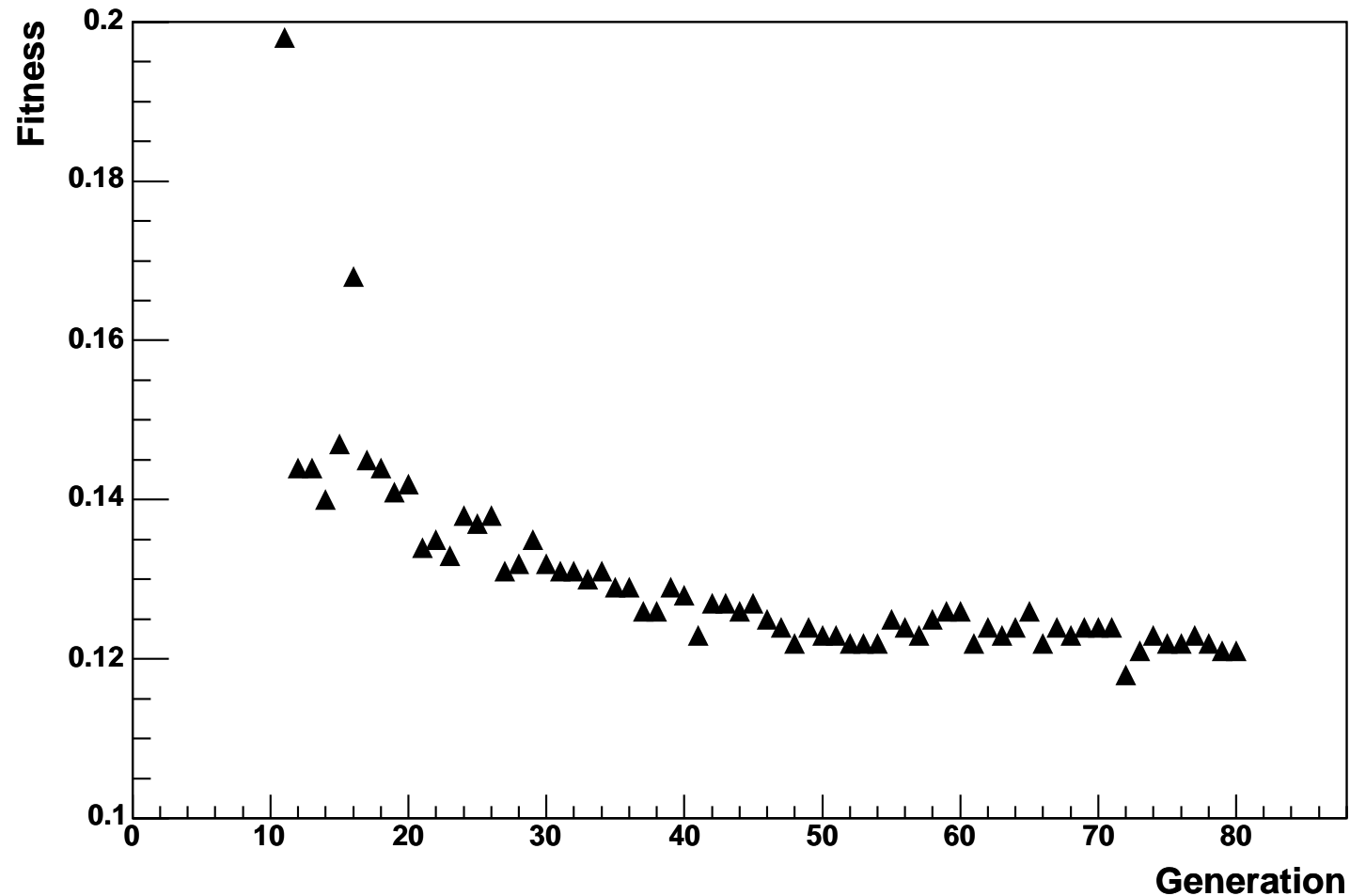
Evolution of Population



Circles: Avg. fitness
Dotted line: Starting fitness

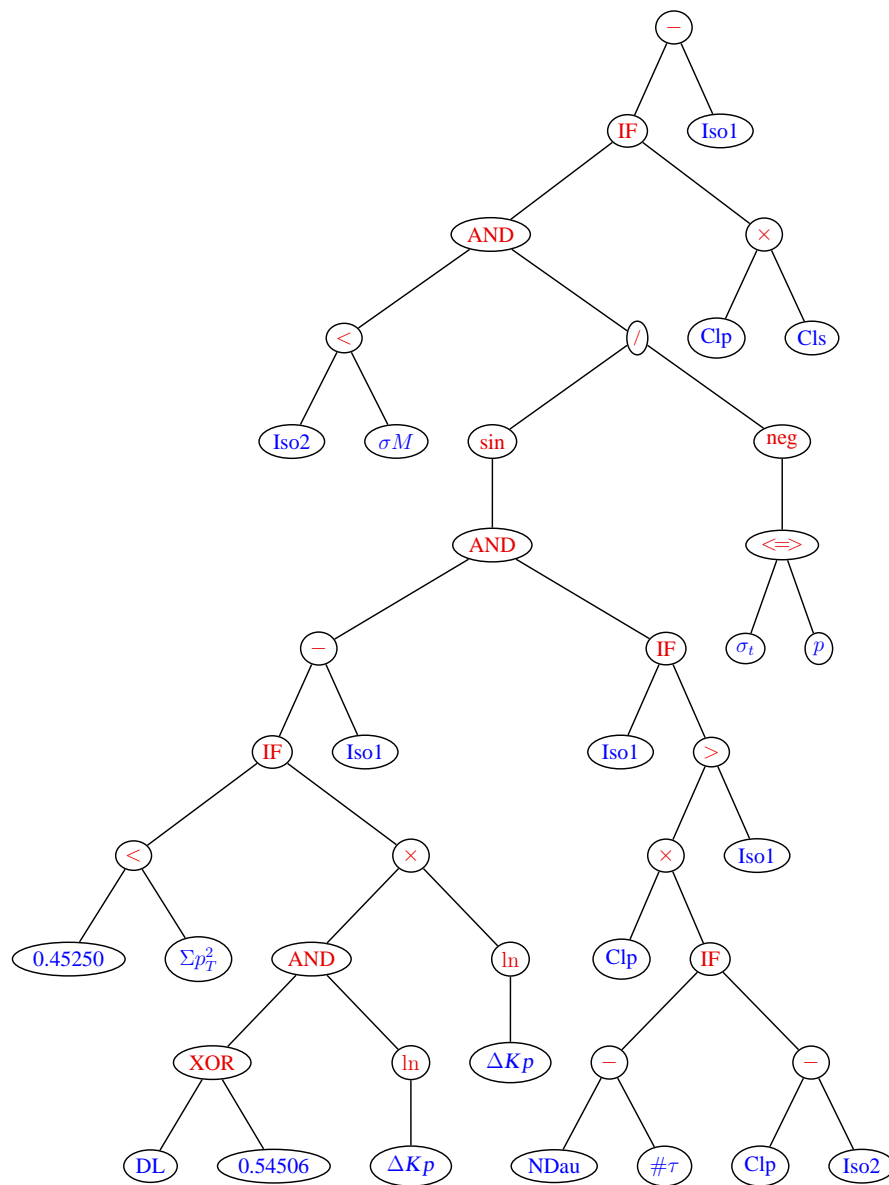
Diamonds: Best Fitness
Solid line: Avg. Size

Evolution of Best Programs



Not much progress after about 50 generations

Best tree (72nd generation)



GP-Induced Systematics

Because this method is new and unsupervised we worry about two things:

- Are the effects of the GP filters well modeled by MC?
- Could it be finding signal where none exists or artificially reducing backgrounds?

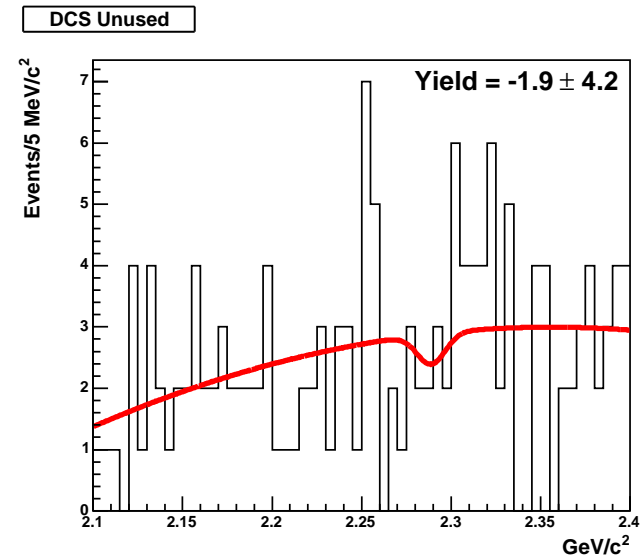
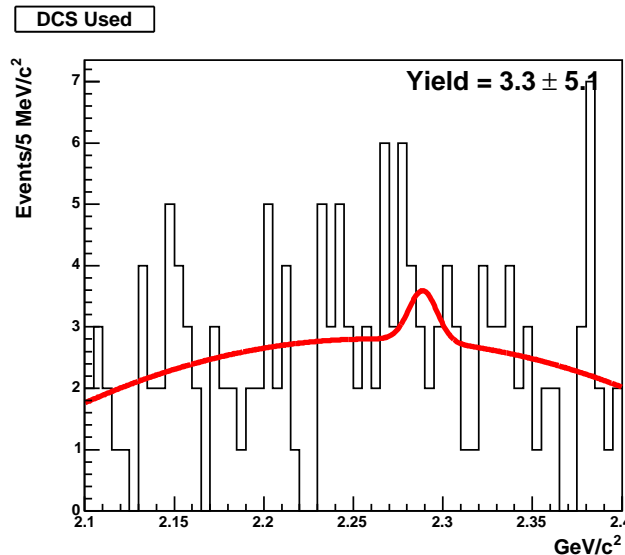
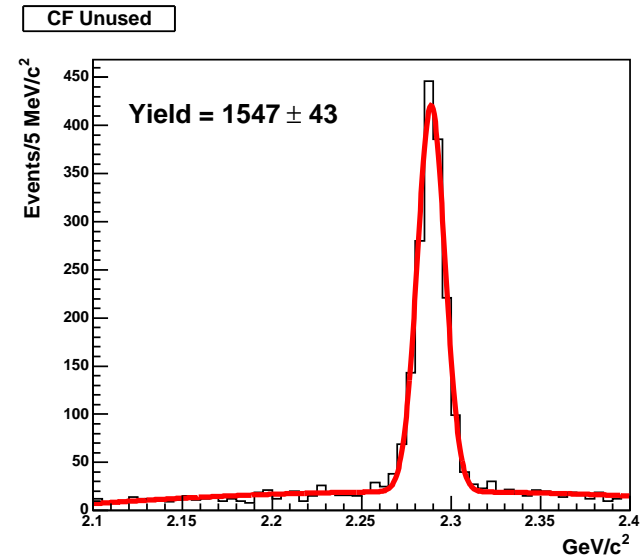
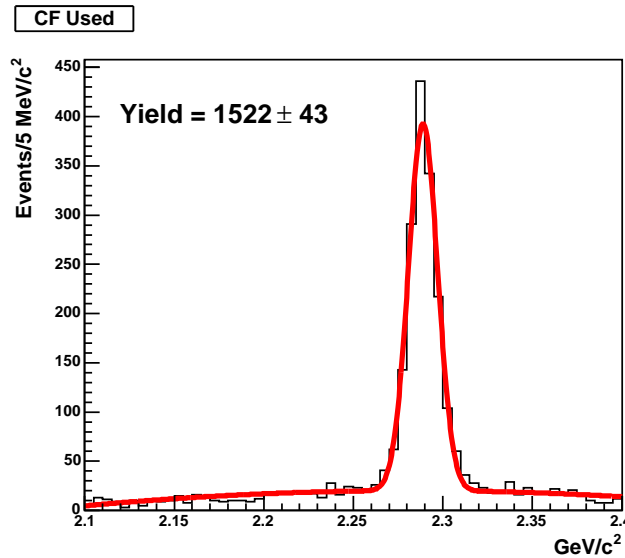
To address the second point, precautions are taken:

- DCS signal region is blinded
- Require a minimum number of events in DCS histogram
- Add a small penalty to fitness for each node
- Only optimize with even numbered events

Last point allows us to check “blindness.” Will also address modeling.

GP Bias Check

No significant discrepancies between used and unused events



GP Efficiency Cross-check

Want to understand if MC accurately predicts efficiency ratio for GP selected events: $\epsilon_{GP}^{DCS} / \epsilon_{GP}^{CF}$. If the MC makes the same mistakes in DCS and CF, this ratio is unaffected.

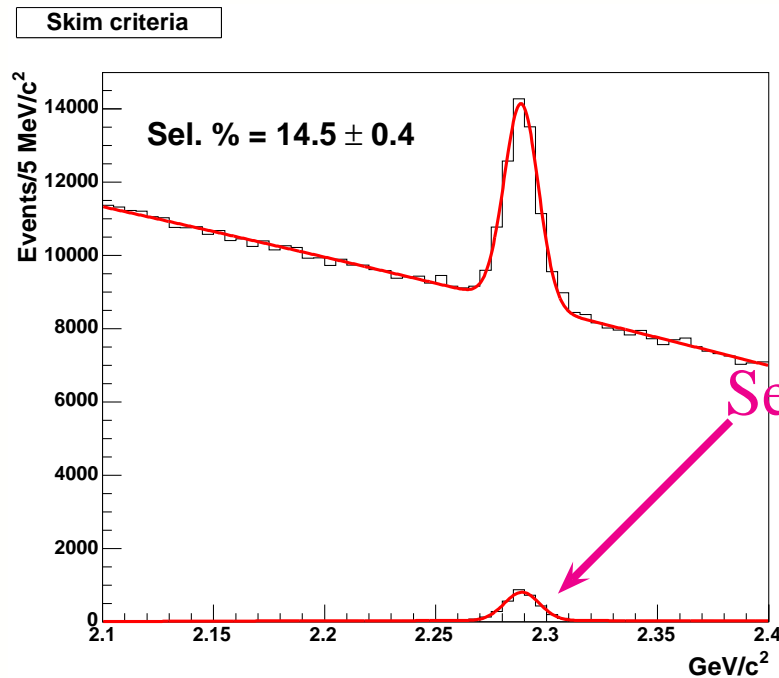
But we cannot see an initial DCS signal, so as a very conservative guess, assume the difference is the same as

$$\frac{\epsilon_{GP}}{\epsilon_{Initial}} \Big|_{CF-Data} \quad \text{compared with} \quad \frac{\epsilon_{GP}}{\epsilon_{Initial}} \Big|_{CF-MC}$$

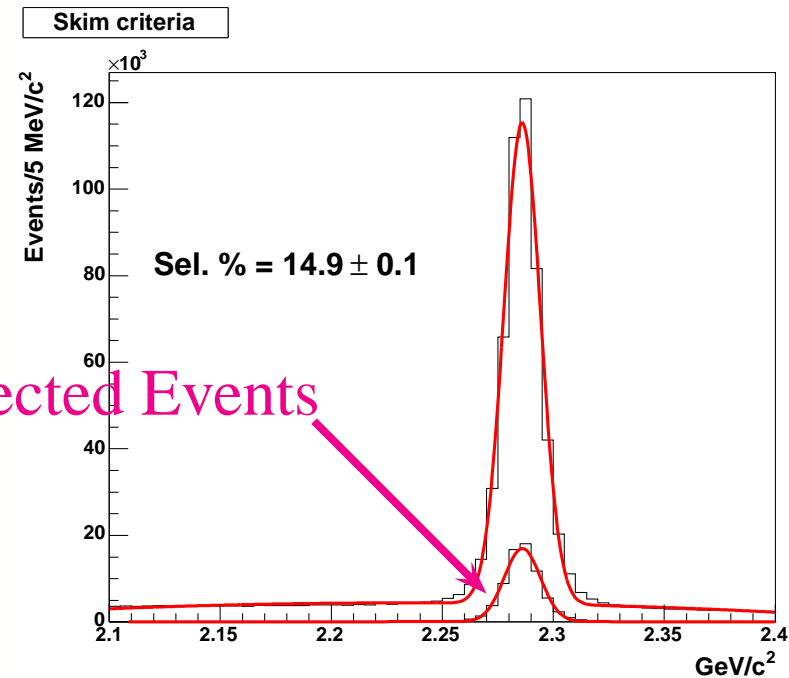
which we can measure.

This ratio can only be right if the MC really models what the GP is doing to the data. (Many studies confirm initial selection ϵ is accurate.)

GP Efficiency Cross-check



Data



Monte Carlo

Within errors, the CF selection efficiencies are the same.

Discrepancy is $2.6 \pm 2.6\%$. I include a 2.6% error on our knowledge of $\epsilon_{\text{DCS}}/\epsilon_{\text{CF}}$.

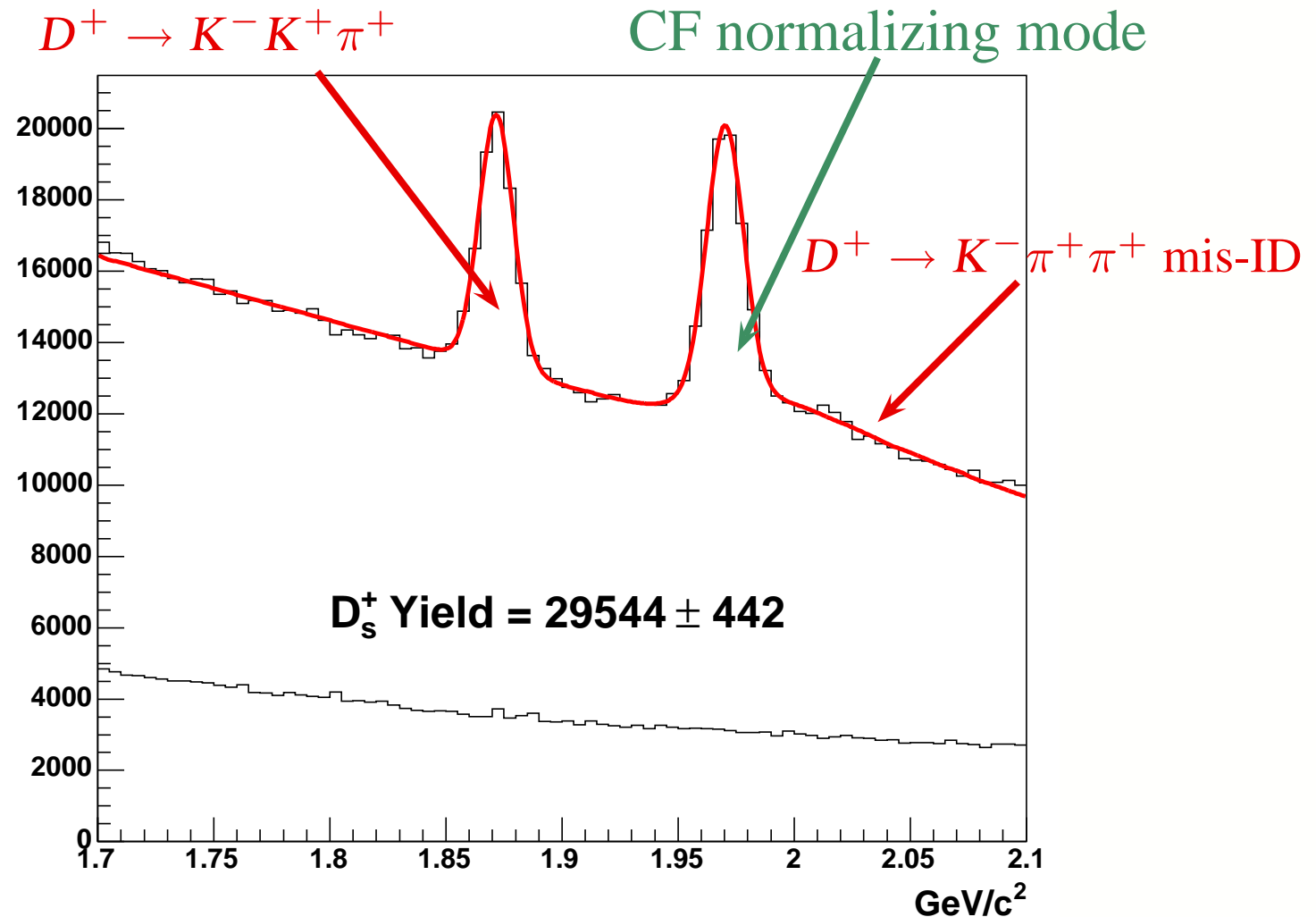
Other Systematic Uncertainties

In addition to GP specific sources of systematic error, consider other sources of uncertainty on our knowledge of $\epsilon_{\text{DCS}}/\epsilon_{\text{CF}}$.

1. Uncertainties in resonance structure of $\Lambda_c^+ \rightarrow pK^-\pi^+$
 - Calculate with world average incoherent and a simpler model with no $\Lambda(1520)^0\pi^+$
 - RMS is 2.1%
2. Uncertainties in resonance structure of $\Lambda_c^+ \rightarrow pK^+\pi^-$.
RMS of
 - 100% non-resonant, 100% $\Delta(1232)^0K^+$, and 100% $pK^*(892)^0$
 - RMS is 5.3%

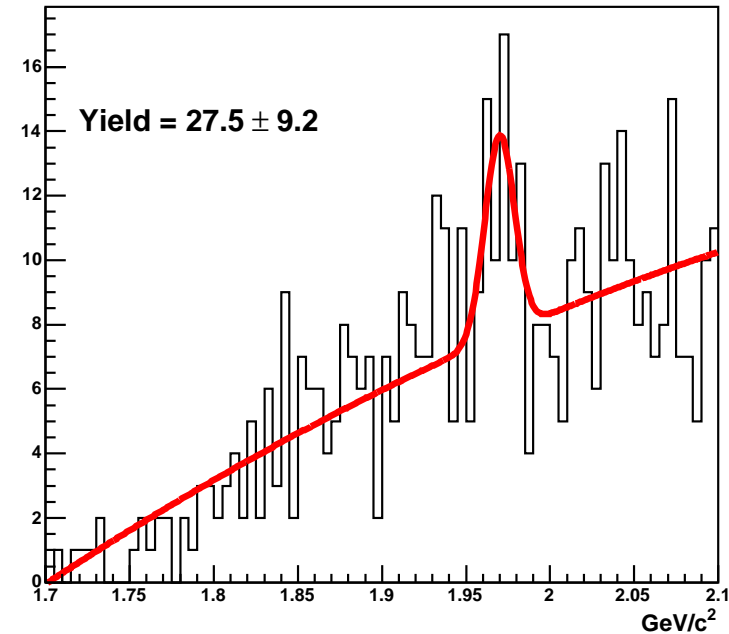
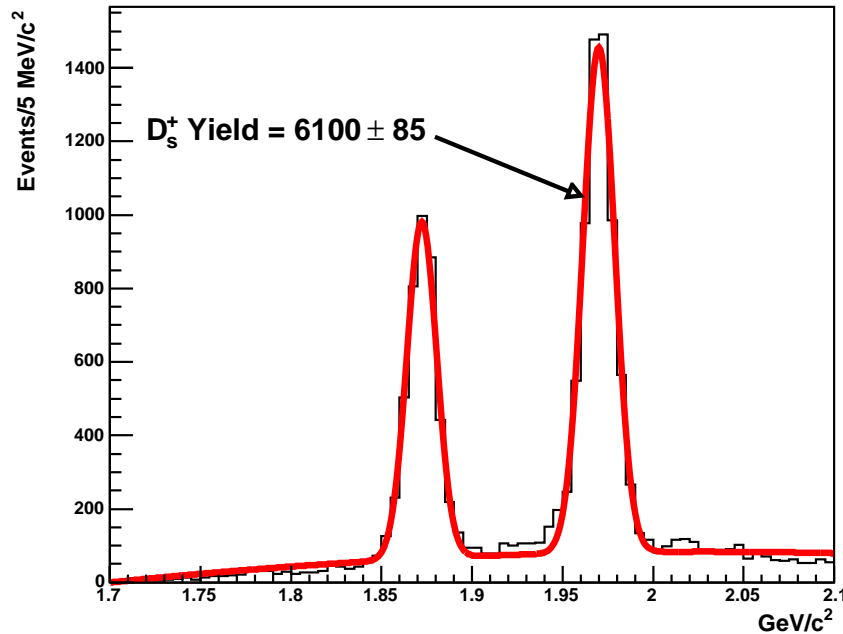
D_s^+ Initial Sample

Contributions from $D^+ \rightarrow K^- \pi^+ \pi^+$ (mis-ID) and Cabibbo suppressed decay $D^+ \rightarrow K^- K^+ \pi^+$ included.



Signals after GP selection

$D^+ \rightarrow K^- \pi^+ \pi^+$ (mis-ID) events removed



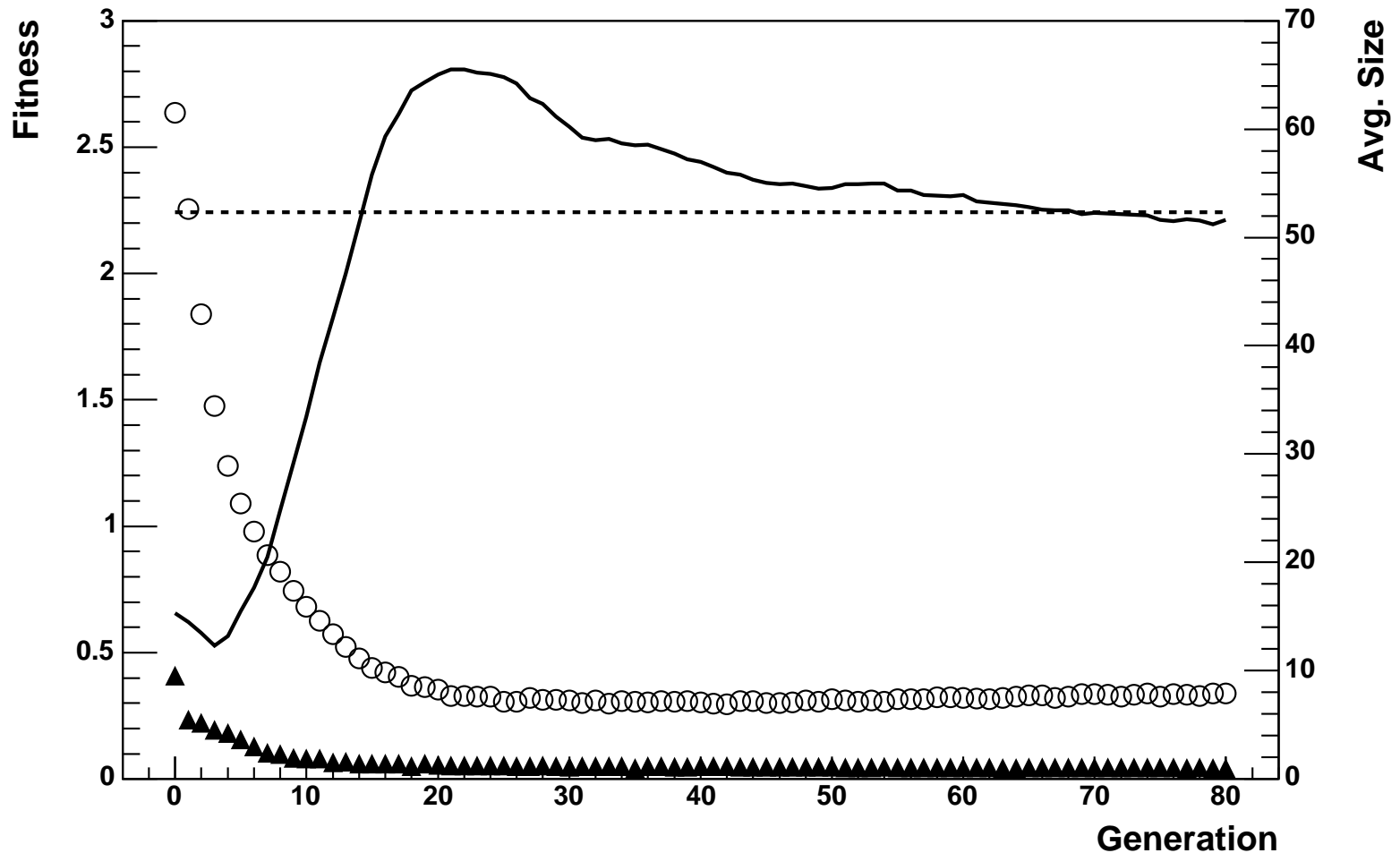
GP retains 21% of original events, DCS BG reduced $\sim 500\times$

Optimization uses 1st degree polynomial, analysis 2nd deg.

No real difference (27.9 ± 9.3 events for 1st deg.)

Same systematic tests as Λ_c^+ , same results

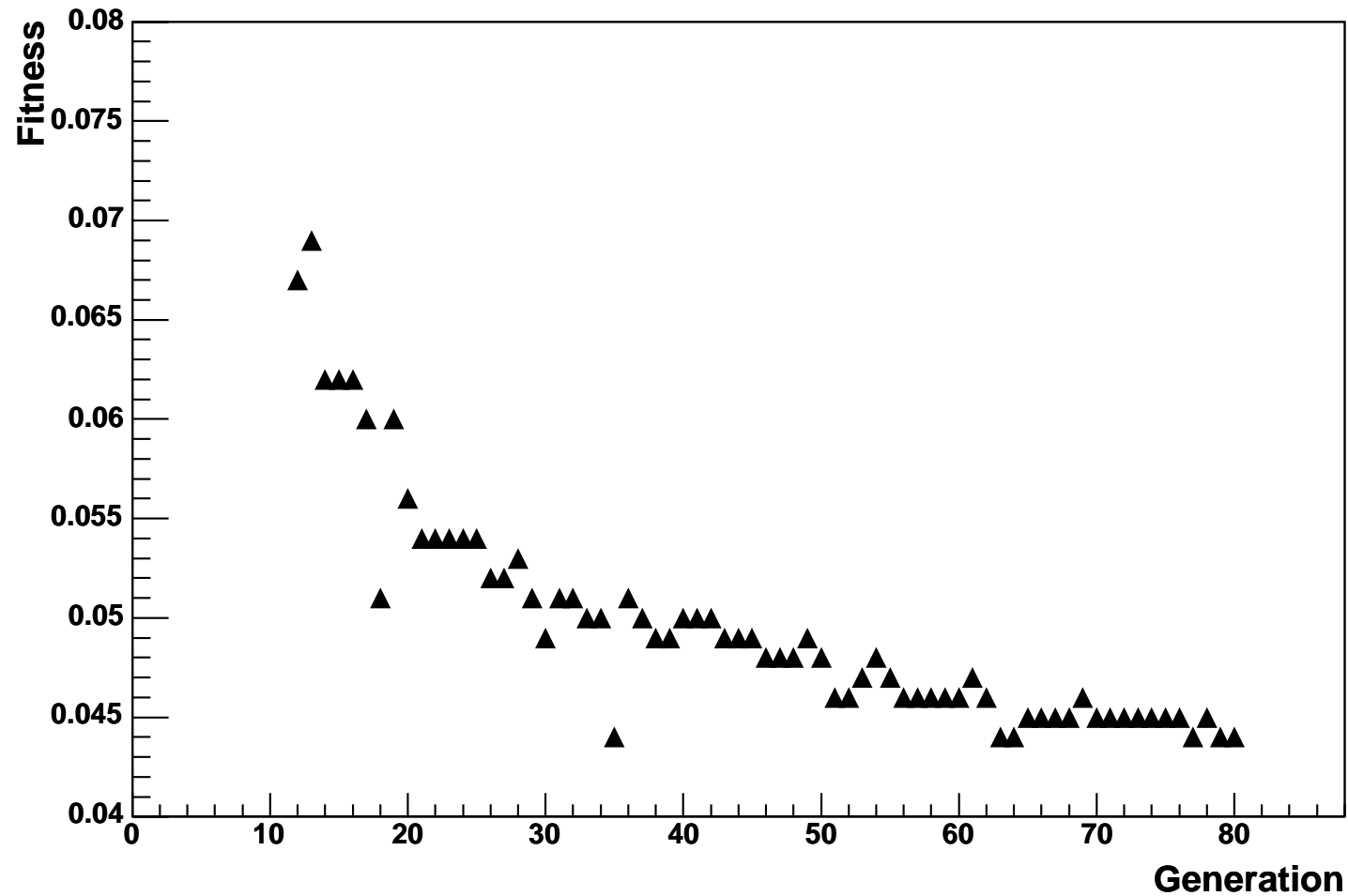
Evolution of Population



Circles: Avg. fitness
Dotted line: Starting fitness

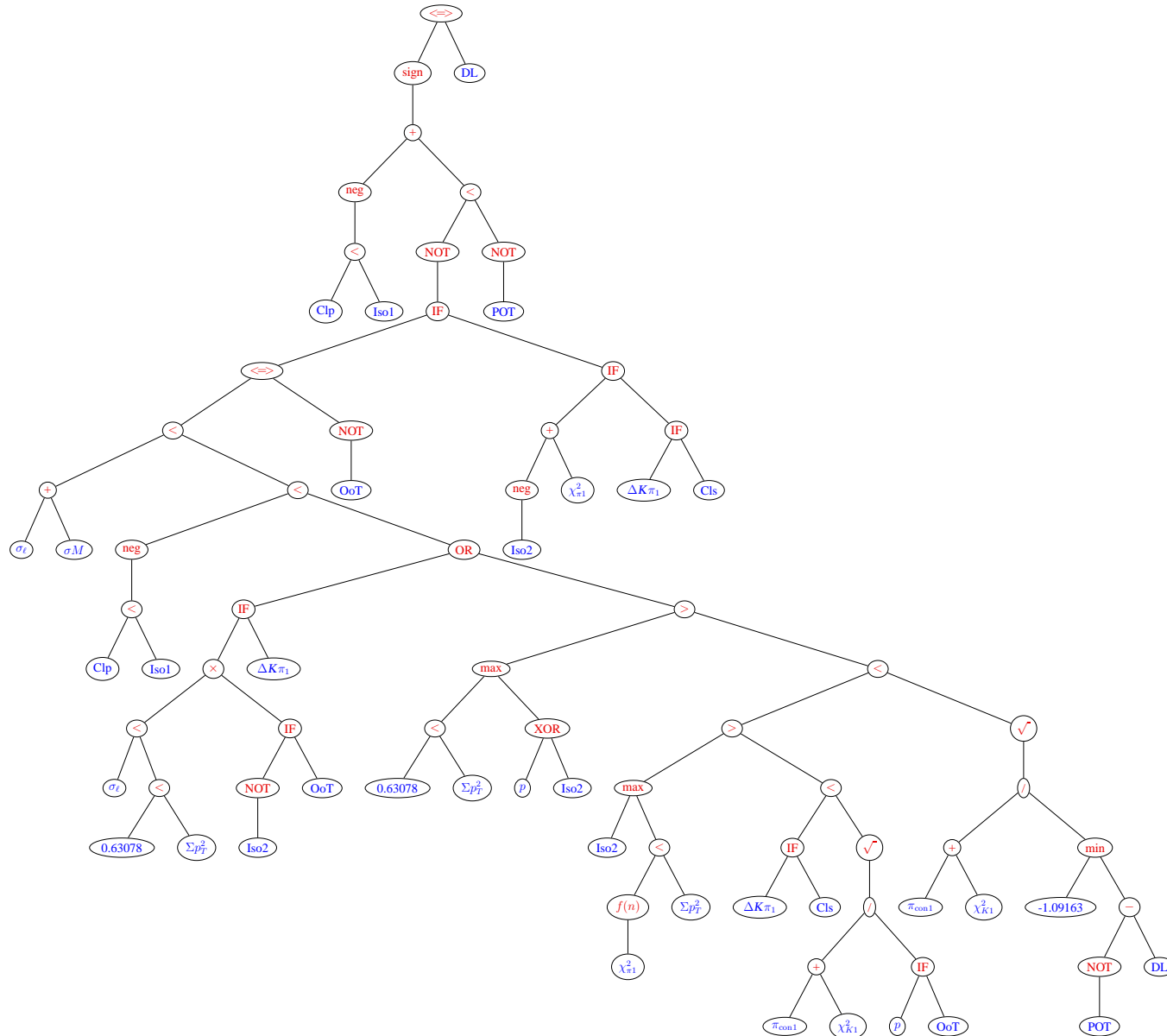
Diamonds: Best Fitness
Solid line: Avg. Size

Evolution of Best Programs



Minor progress right up to end

Best tree (63rd generation)



D_s^+ Systematic Uncertainties

Same tests as for Λ_c^+ , similar results.

1. Uncertainties in resonance structure of $D_s^+ \rightarrow K^- K^+ \pi^+$
 - Calculate with world average incoherent and a FOCUS coherent model
 - RMS is 2.6%
2. Uncertainties in resonance structure of $D_s^+ \rightarrow K^+ K^+ \pi^-$.
RMS of
 - 100% non-resonant, 100% $K_0^*(1430)^0 K^+$, and 100% $K^*(892)^0 K^+$
 - RMS is 10.7%

Systematic Uncertainties

We tabulate and add in quadrature all uncertainties on relative reconstruction efficiency, $\epsilon_{\text{DCS}}/\epsilon_{\text{CF}}$:

Source	Syst. Unc. (%)	
	Λ_c^+	D_s^+
MC statistics	0.6	0.4
DCS resonances	5.3	10.7
CF resonances	2.1	2.6
GP filter	2.6	3.5
Total	6.3	11.6

Including systematics

Often systematic uncertainties on limit measurements are ignored. Instead, we use our % uncertainty in our knowledge of the DCS vs. CF efficiencies and use a technique from Convery (SLAC-TN-03-001) to calculate systematic errors. This method builds on the Feldman-Cousins method and can be applied to situations with more background.

In this case we get a BR probability of

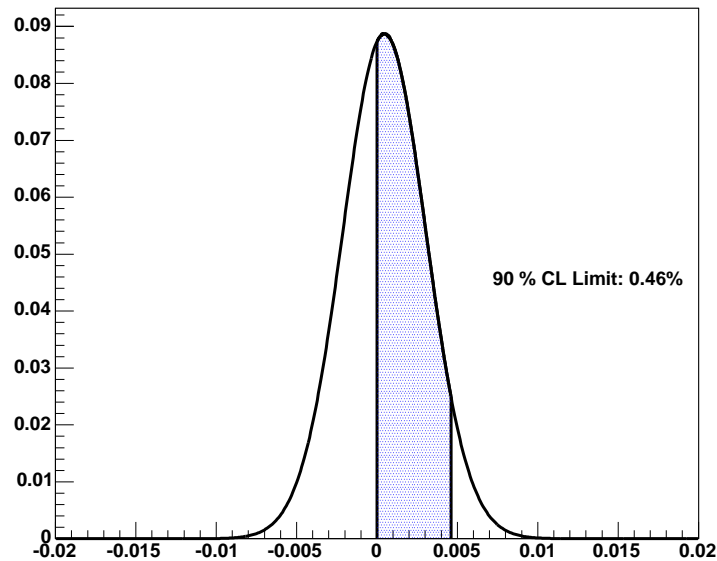
$$p(B) \propto \frac{1}{\sqrt{\frac{B^2}{2\sigma_B^2} + \frac{\hat{S}^2}{2\sigma_S^2}}} e^{-\frac{(B-\hat{B})^2}{2\left(\frac{B^2\sigma_S^2}{\hat{S}^2} + \sigma_B^2\right)}}$$

where \hat{B} is nominal BR, σ_B is its error and σ_S/\hat{S} is the percent error on our knowledge of the efficiency. Assumes $\sigma_S/\hat{S} \lesssim 0.25$.

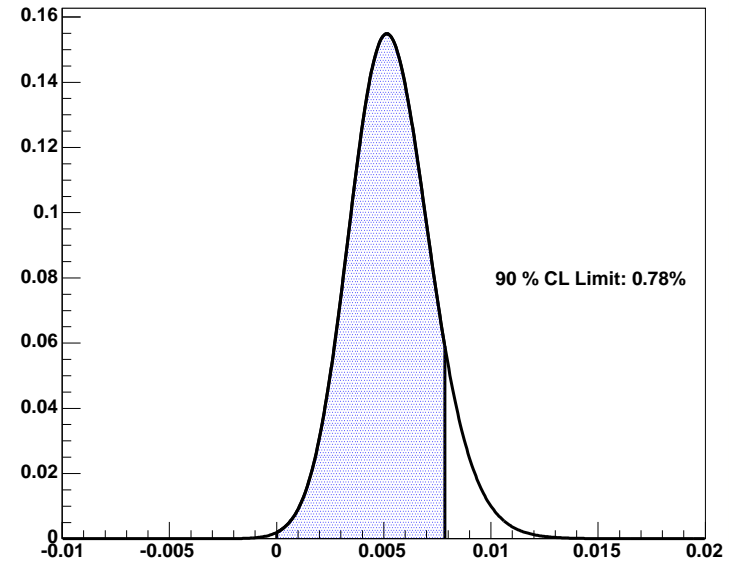
Limit Determination

We then integrate $P(B)$ over the physical range to find 90% coverage/limit.

BR Probability



$$P(B) \text{ for } \Lambda_c^+ \rightarrow p K^+ \pi^-$$



$$P(B) \text{ for } D_s^+ \rightarrow K^+ K^+ \pi^-$$

Summary of measurements

With this method, we can express our measurements as either central values with errors or limits:

Decay mode	Central Value	Limit (90% CL)
$\frac{\text{BR}(\Lambda_c^+ \rightarrow p K^+ \pi^-)}{\text{BR}(\Lambda_c^+ \rightarrow p K^- \pi^+)}$	$(0.05 \pm 0.26 \pm 0.02)\%$	$< 0.46\%$
$\frac{\text{BR}(D_s^+ \rightarrow K^+ K^+ \pi^-)}{\text{BR}(D_s^+ \rightarrow K^- K^+ \pi^+)}$	$(0.52 \pm 0.17 \pm 0.11)\%$	$< 0.78\%$

- First limits for these decays (or any DCSD of D_s^+ or Λ_c^+)
- Approach sensitivity of expected values $\sim \tan^4 \theta_c = 0.25\%$
- There is a hint of a signal in $D_s^+ \rightarrow K^+ K^+ \pi^-$ which would not fit with naïve theory
 - Because the decay $D^+ \rightarrow K^+ \pi^+ \pi^-$ has a Rel. BR $\approx 3 \tan^4 \theta_c$, $D_s^+ \rightarrow K^+ K^+ \pi^-$ should be $\approx \frac{1}{3} \tan^4 \theta_c$

Conclusions (Analysis)

- These are the first limits for DCS decays of D_s^+ and Λ_c^+
- Efficiency of GP event selection is well modeled in FOCUS
- I've published two articles on this subject
 - NIM article on method, applied to $D^+ \rightarrow K^+ \pi^+ \pi^-$:
hep-ex/0503007, NIMA 551, pg. 504
 - D_s^+ and Λ_c^+ rel. branching ratio results:
hep-ex/0507103, PLB 624, pg. 166

Conclusions (GP Method)

- I hope I've been able to give you a flavor of Genetic Programming.
- We have shown that GP can be used in HEP event selection (this is the first application on HEP data).
- Can be used to improve sensitivity over traditional techniques (factors of $\sqrt{2}$ are possible).
- Trees are very complex and any attempt to understand the whole thing may be pointless.
 - What it is doing is visible. Why it works may not be.
 - Can be changed and disassembled into parts.
 - One combination which appeared often is now being used in other analyses.
- I think this novel method deserves further exploration.
Thank you for your attention.