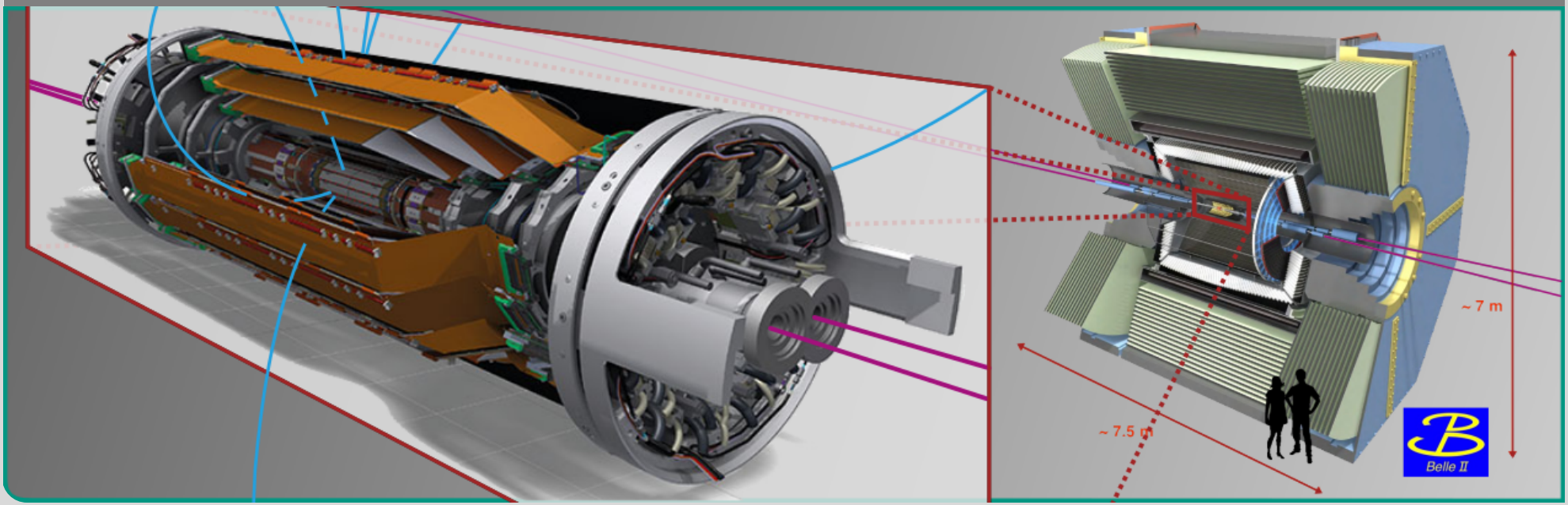
Karlsruhe Institute of Technology

VXDTF2 Modified Candidate Selection

Jonas Wagner | 08.09.2017

KIT - INSTITUT FÜR EXPERIMENTELLE TEILCHENPHYSIK



Current approach

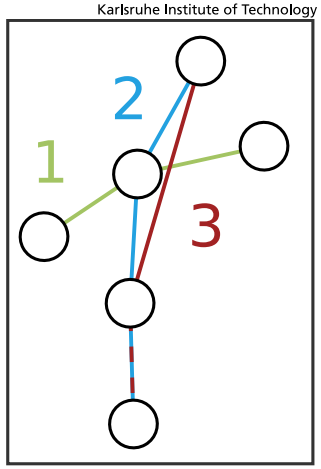
- Candidate creation
- Quality estimation
- OverlapMatrix creation – Which tracks share clusters
- Greedy/Hopfield – No shared clusters allowed

Efficiency loss

- 1-2 % loss in finding efficiency due to MC tracks sharing clusters

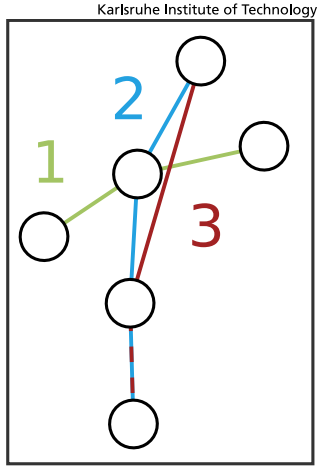
How to prevent efficiency loss?

- Two track candidates share spacepoints
- Instead of deactivating the competing candidate, only remove spacepoint
- Which candidate should keep the spacepoint?
- How to incorporate this into Greedy/Hopfield?



How to prevent efficiency loss?

- Two track candidates share spacepoints
- Instead of deactivating the competing candidate, only remove spacepoint
- Which candidate should keep the spacepoint?
- How to incorporate this into Greedy/Hopfield?

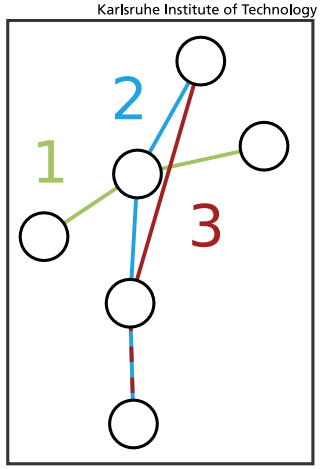


Chosen approach

- Add candidates that don't include the spacepoint
- Estimate quality of additional candidates
- \+ Greedy/Hopfield can be used directly
- \- A lot more candidates

How to prevent efficiency loss?

- Two track candidates share spacepoints
- Instead of deactivating the competing candidate, only remove spacepoint
- Which candidate should keep the spacepoint?
- How to incorporate this into Greedy/Hopfield?



Chosen approach

- Add candidates that don't include the spacepoint
- Estimate quality of additional candidates
- \+ Greedy/Hopfield can be used directly
- \- A lot more candidates

Possible Problem

- Could have negative influence on fake rate

How to reduce number of candidates?

Resolve overlap in two steps:

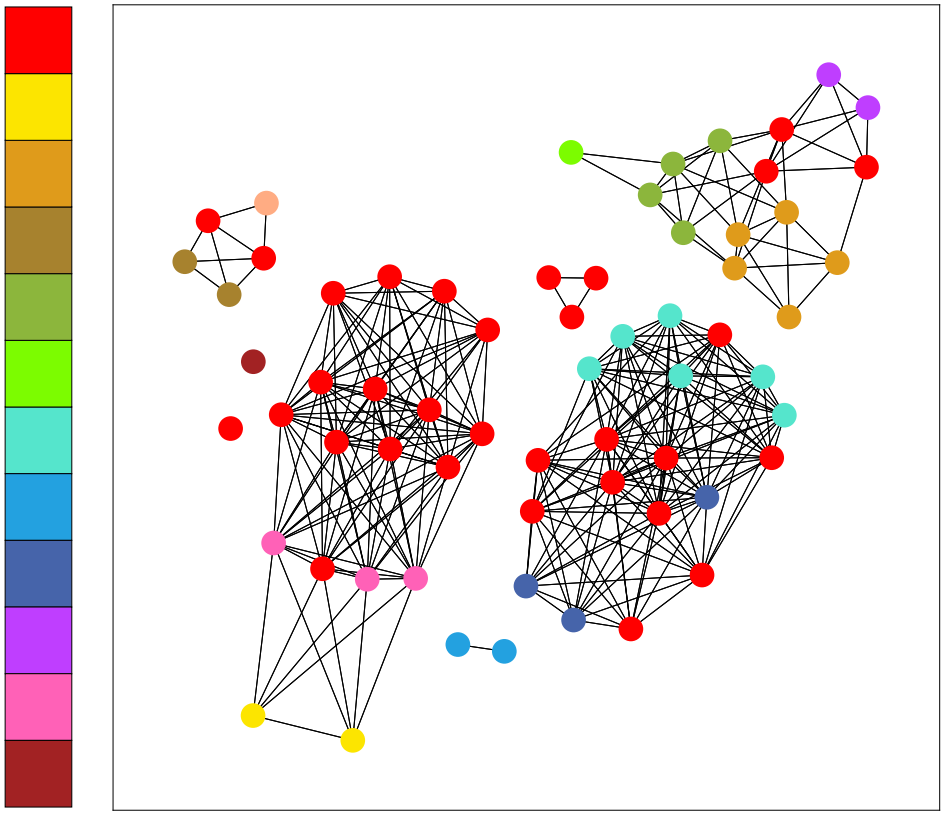
First step

- Two tracks are competing only if they share at least two spacepoints
- All competing track candidates are part of a family

$\Rightarrow$ Resolve overlap for each familiy separately

$\Rightarrow$ Bonus: smaller overlap matrices!

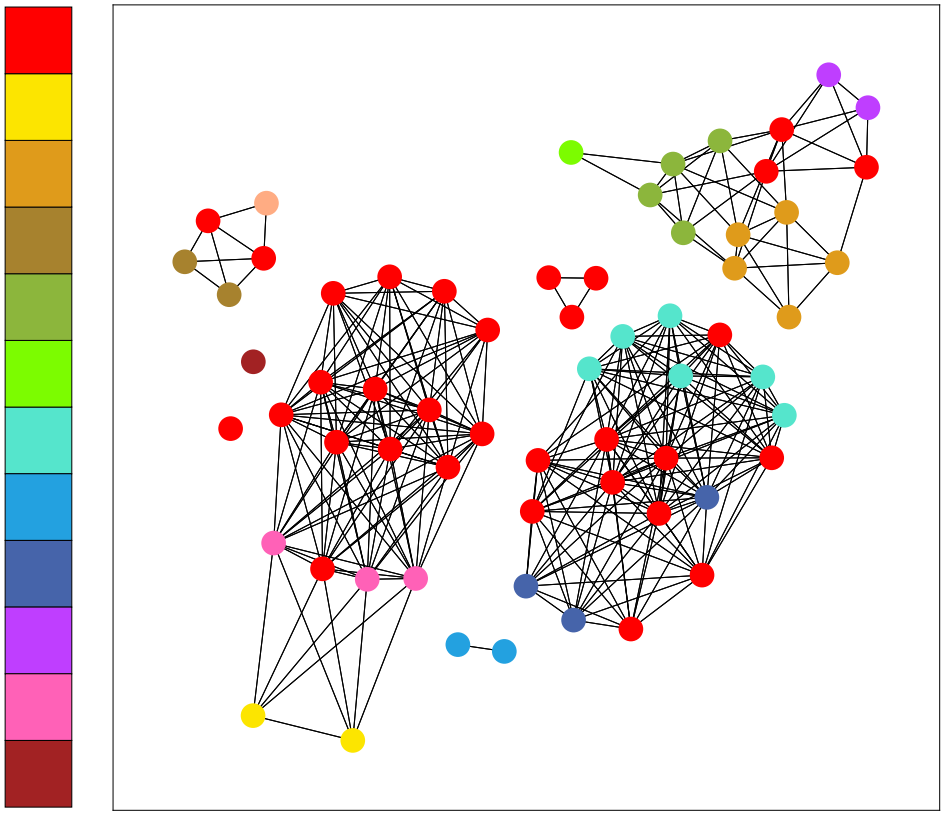
Example Event – Step 1



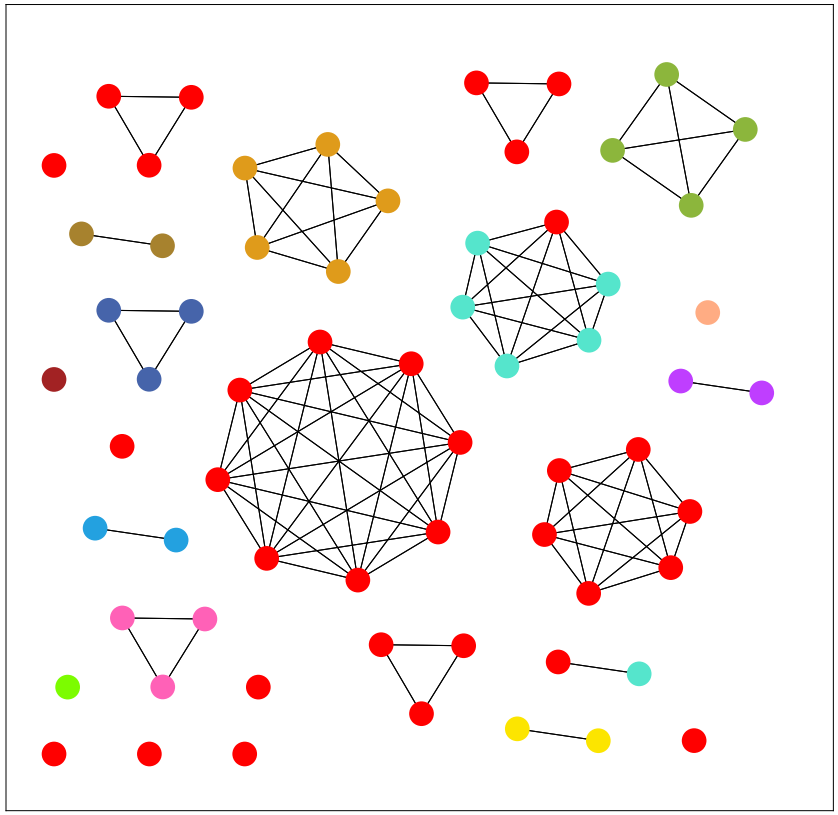
Cluster overlap

- Each node represents one candidate
- Two nodes are connected if they are overlapping
- The colour represents the matched MC track
- Bright red represents no match, a fake

Example Event – Step 1



Cluster overlap



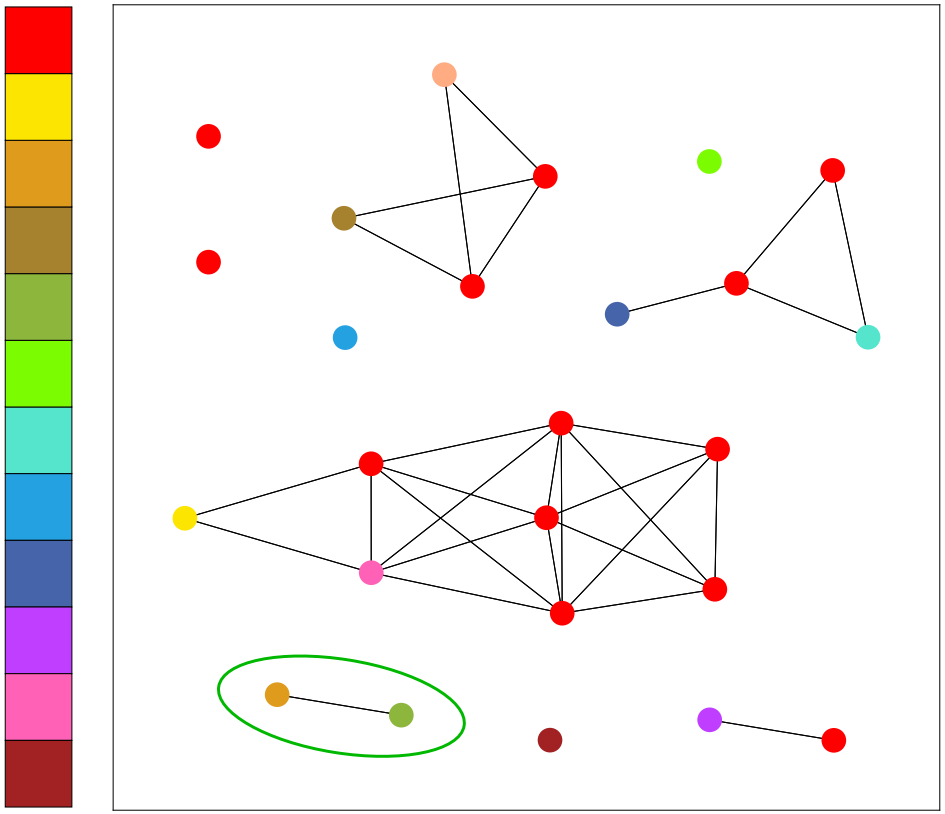
2 spacepoint overlap

Step 1 – Figures of Merit

	All candidates	Best candidates
Average clones	31.35 ± 0.05	2.81 ± 0.02
Average fakes	294.41 ± 0.06	15.18 ± 0.03
Finding efficiency	96.24 ± 0.06	95.94 ± 0.06
Hit efficiency	97.13 ± 0.03	97.11 ± 0.03
Hit purity	83.78 ± 0.02	93.02 ± 0.04

- Numbers based on MCQualityEstimator

Example Event – After Step 1



2 spacepoint overlap resolved
Cluster overlap displayed

2 Step candidate selection

Resolve overlap in two steps:

First step

- Two tracks are competing only if they share at least two spacepoints
- All competing track candidates are part of a family

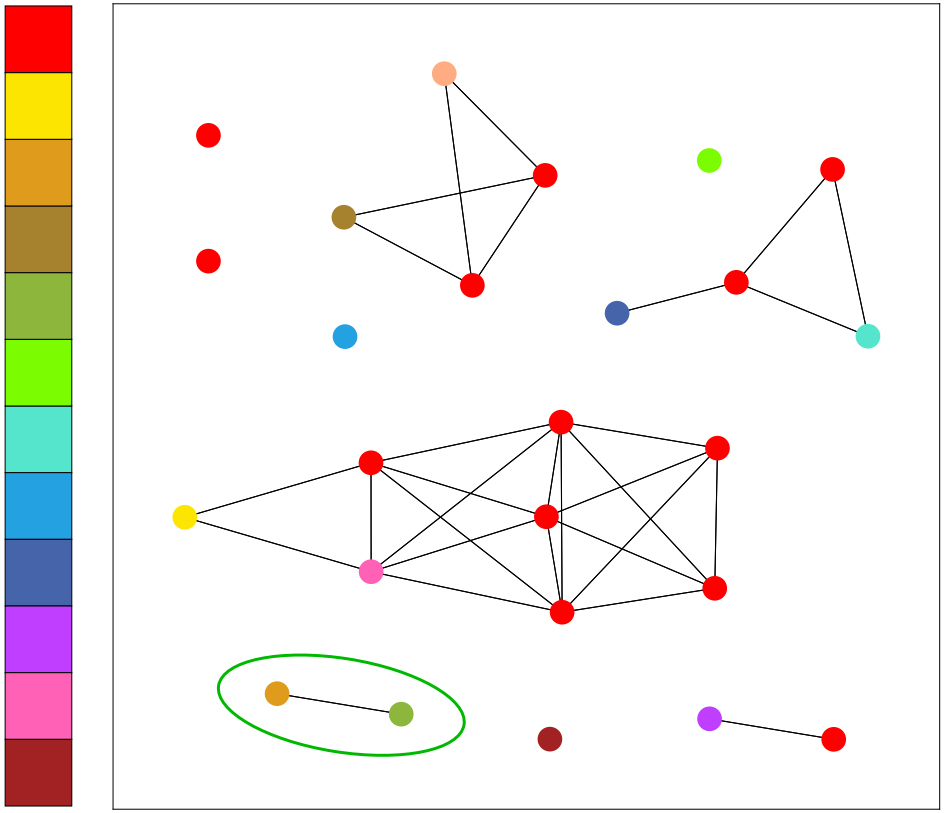
$\Rightarrow$ Resolve overlap for each familiy separately

$\Rightarrow$ Bonus: smaller overlap matrices!

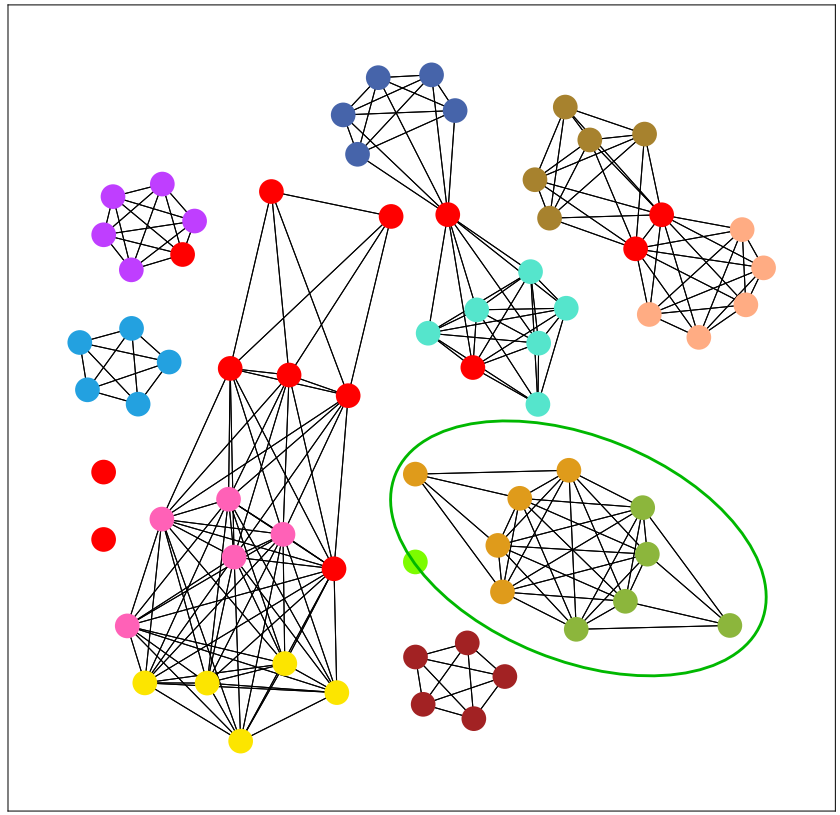
Second step

- Add subtracks
- Estimate quality
- Resolve cluster overlap

Example Event – Step 2



2 spacepoint overlap resolved
Cluster overlap displayed



Cluster overlap including subtracks

Results – Figures of Merit

Quality estimator	MCQualityEstimator		TripletFit	
Selection method	Improved	Original	Improved	Original
Clone rate	0.40 ± 0.02	0.41 ± 0.02	0.40 ± 0.02	0.43 ± 0.02
Fake rate	18.56 ± 0.11	19.81 ± 0.12	20.03 ± 0.12	20.62 ± 0.12
Finding efficiency	95.37 ± 0.07	94.48 ± 0.07	94.01 ± 0.08	93.58 ± 0.08
Hit efficiency	96.97 ± 0.03	97.21 ± 0.03	95.36 ± 0.04	96.36 ± 0.04
Hit purity	99.87 ± 0.01	99.76 ± 0.01	98.95 ± 0.01	98.85 ± 0.01

- MCQE overall better results, but slight decrease of hit efficiency

$\Rightarrow$ Expected, because spacepoints are removed

- Smaller improvements for Tripletfit and worse hit efficiency loss

$\Rightarrow$ Imperfect quality estimation

Results – Runtime?

Additional steps:

- 2 spacepoint overlap determined in SegmentNetwork

$\Rightarrow$ All paths that share a node, are in the same family

- Family is stored in SPTC
- BestFamiliyCandidateSelection is indistinguishable from Greedy selection in the first step
- Add candidates
- Estimate quality
- Second step overlap resolving as usual

Results – Runtime!

Average time spent per event

		Original ms	Improved ms
Candidate Creation	Spacepoint Creation	1.7	1.7
	Network Creation	8.1	8.2
	Path Finding	2.1	2.2
Quality Estimation		0.5	0.6
Candidate Selection		3.9	1.8
		16.3	14.5

Warning – Rough measurement!

- Performed on my desktop at idle times
- 10k events Y4S + BKG

Could it be even better?

	All candidates	Best candidates
Average clones	31.35 ± 0.05	2.81 ± 0.02
Average fakes	294.41 ± 0.06	15.18 ± 0.03
Finding efficiency	96.24 ± 0.06	95.94 ± 0.06
Hit efficiency	97.13 ± 0.03	97.11 ± 0.03
Hit purity	83.78 ± 0.02	93.02 ± 0.04

Could it be even better?

	All candidates	Best candidates
Average clones	31.35 ± 0.05	2.81 ± 0.02
Average fakes	294.41 ± 0.06	15.18 ± 0.03
Finding efficiency	96.24 ± 0.06	95.94 ± 0.06
Hit efficiency	97.13 ± 0.03	97.11 ± 0.03
Hit purity	83.78 ± 0.02	93.02 ± 0.04

Implement first step during path finding

- Same results as shown before
- Only a fraction of SPTCs

$\Rightarrow$ Overall faster

$\Rightarrow$ Memory consumption of VXDTF2 reduced by a large fraction!

Even better runtime

		Original ms	Improved ms	Even Better ms
Candidate Creation	Spacepoint Creation	1.7	1.7	1.6
	Network Creation	8.1	8.2	8.1
	Path Finding	2.1	2.2	0.7
Quality Estimation		0.5	0.6	0.2
Candidate Selection		3.9	1.8	1.2
		16.3	14.5	11.7

Further advantage

No direct memory influence of:

- Additional paths from SectorMap
- Additional paths from BasicPathFinder
- Subtracks during the candidate creation

Overview – Smaller, Faster, Better

Advantages	Disadvantages
Better finding efficiency	Lower hit efficiency
Less memory	PXD influence not tested
Faster	Breaks module separation
Additional path finding options	

Backup

Additional subtracks during path finding