

SBND CM

Wire-Cell Event Reconstruction

Part 1: Sim-NP-SigProc

Part 2: Img-Clus-QLMatching-nu-select

Part 3: 3D PatRec



TPC simulation

noise filtering

signal processing

3D imaging

clustering

charge-light matching

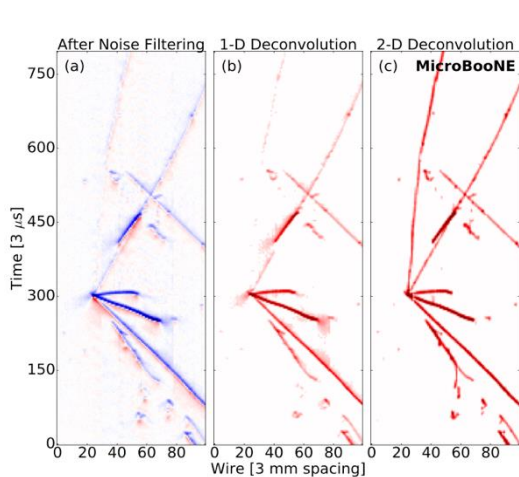
3D trajectory &
dQ/dx fitting

cosmic muon tagger

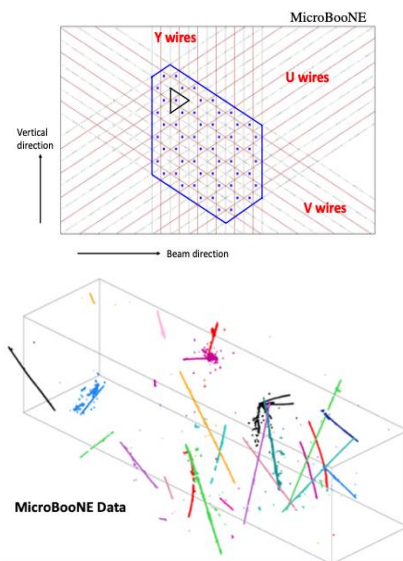
multi-track fitting

DL-3D vertexing

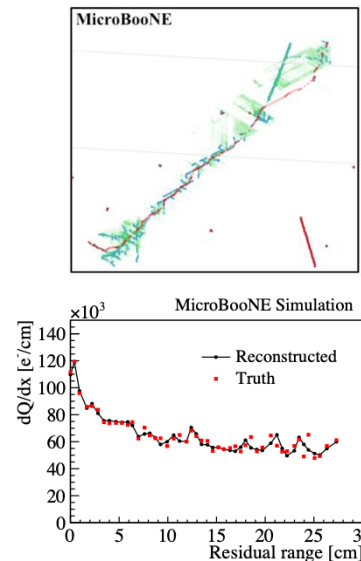
particle identification



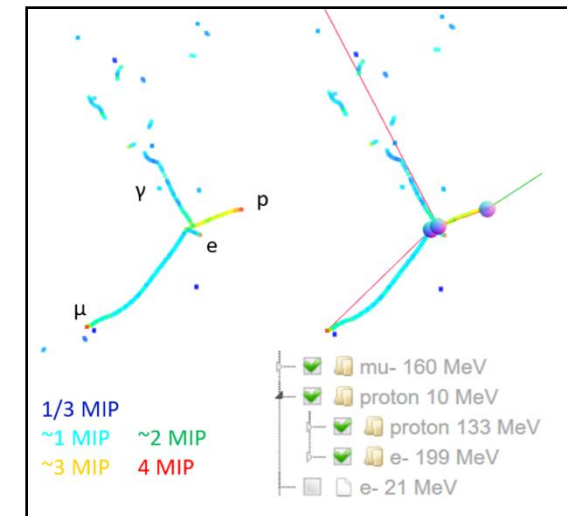
[JINST 12 P08003 \(2017\)](#)
[JINST 13 P07006 \(2018\)](#)
[JINST 13 P07007 \(2018\)](#)
[JINST 16 P01036 \(2020\)](#)



[JINST 13 P05032 \(2018\)](#)
[JINST 16 P06043 \(2021\)](#)



[Phys. Rev. Applied 15, 064071 \(2021\)](#)

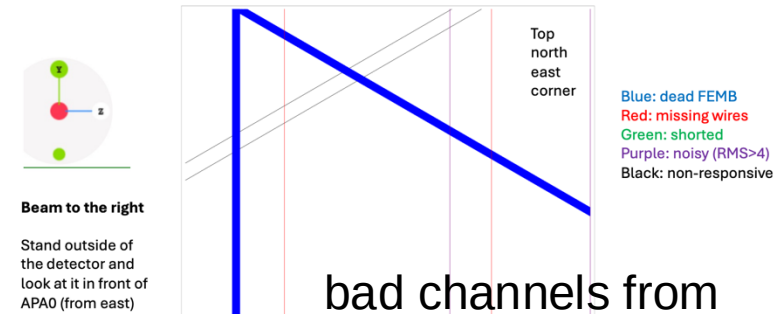


[JINST 17 P01037 \(2022\)](#)

SBND Clustering example

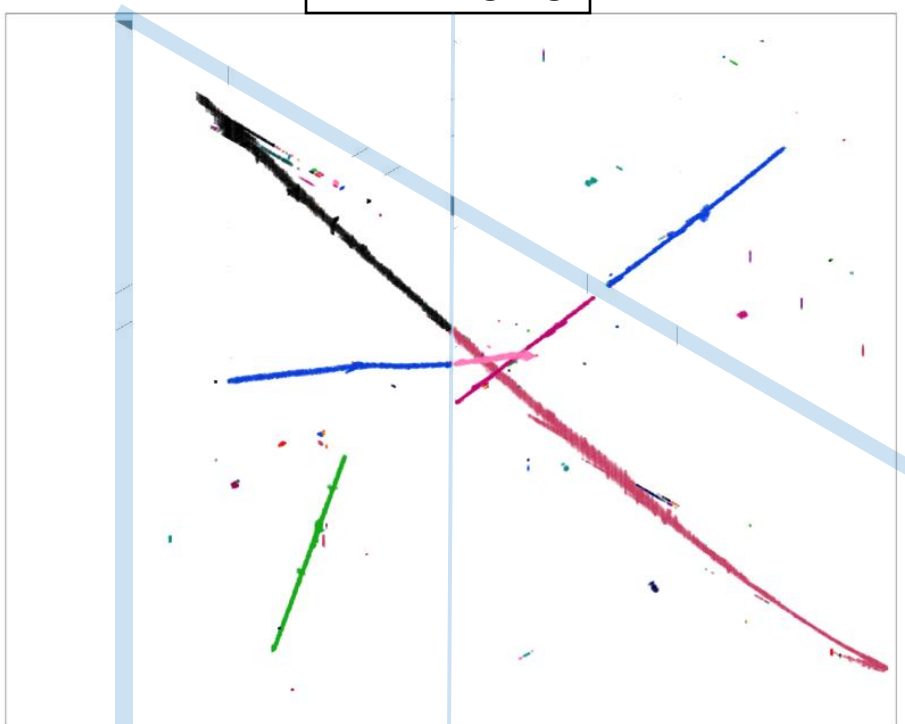
- SBND run 14737, event 279
- jumping gaps due to bad channel and APA center gap
- smaller activities clustered to larger ones

APA0 (East) bad channels [real geometry]



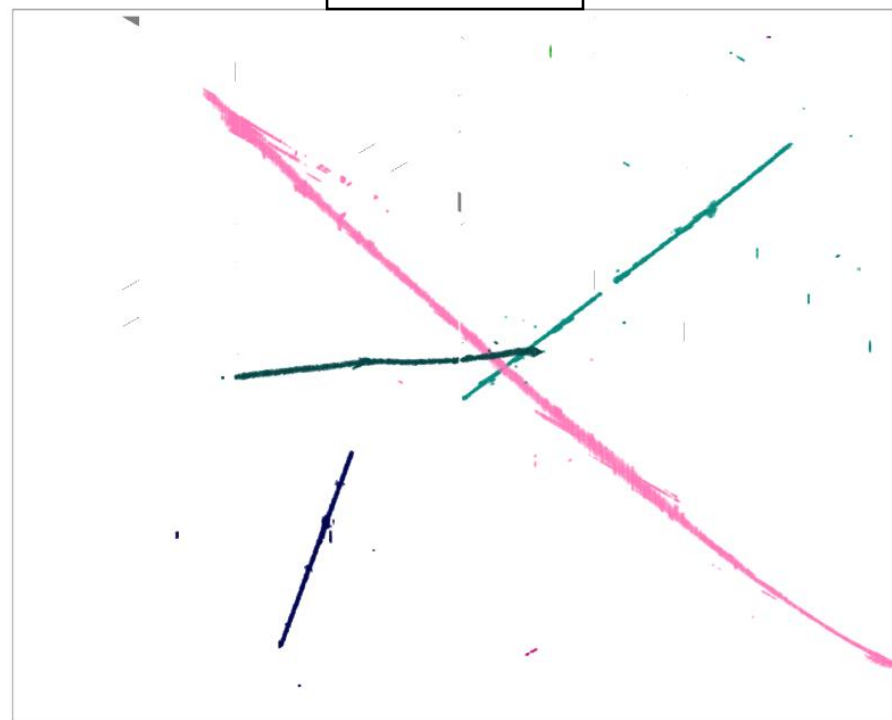
bad channels from
Tom and Hanyu's talk

3D imaging



APA center gap

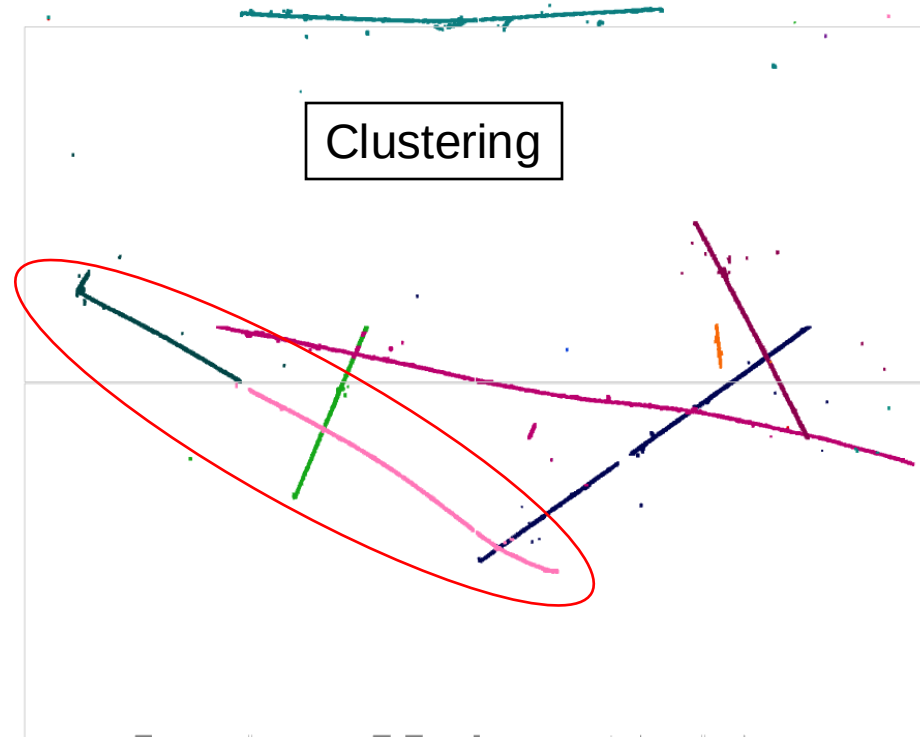
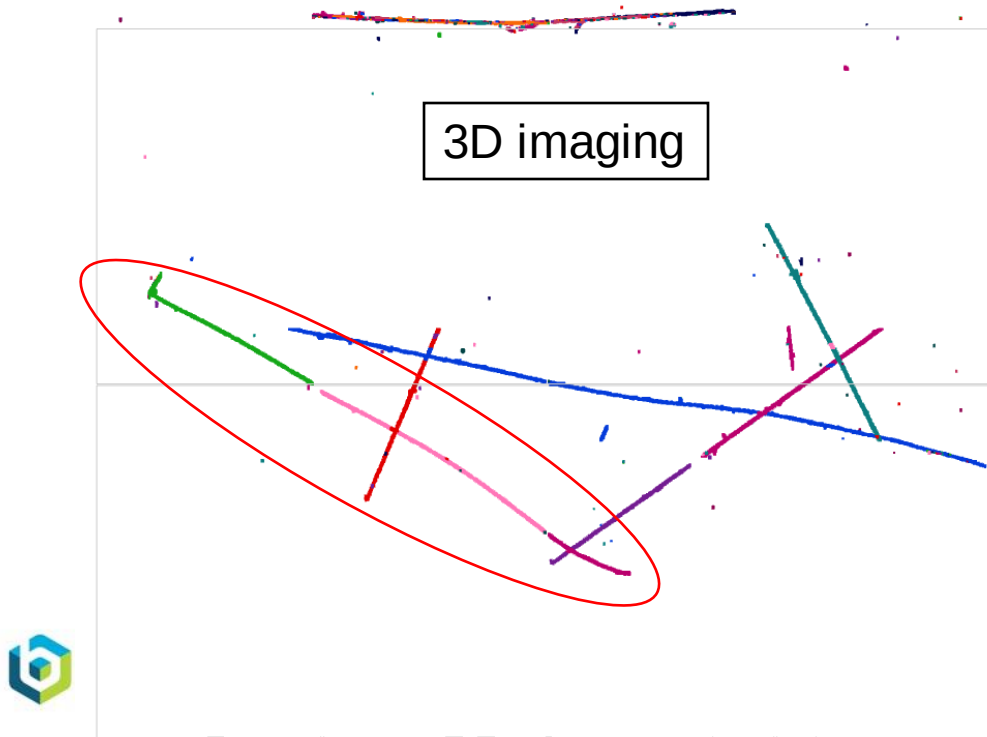
Clustering



<https://www.phy.bnl.gov/twister/bee/set/35591e89-5845-47ad-8214-52412ef7cfa1/event/0/>

Top view of both APAs

- Cross APA clustering underdevelopment
- A list of 4 SBND event: [link](#)



to-do 1

- ✓ fix known bugs in two functions
 - ✓ Brett: new separate() function
 - Xin: validation

```
local mabc = g.pnode({
  type: "MultiAlgBlobClustering",
  name: "%s-%d"%[anode.name, face],
  data: {
    inpath: "pointtrees/%d",
    outpath: "pointtrees/%d",
    perf: true,
    bee_dir: bee_dir, // "data/0/0", // not used
    bee_zip: "mabc-%s-face%d.zip"%[anode.name, face],
    bee_detector: "sbnd",
    initial_index: index, // New RSE configuration
    use_config_rse: true, // Enable use of configured RSE
    runNo: LrunNo,
    subRunNo: LsubRunNo,
    eventNo: LeventNo,
    save_deadarea: true,
    anode: wc.tn(anode),
    face: face,
    geom_helper: wc.tn(geom_helper),
    func_cfgs: [
      {name: "clustering_live_dead", dead_live_overlap_offset: 2},
      {name: "clustering_extend", flag: 4, length_cut: 60 * wc.cm, num_try: 0, length_2_cut: 15 * wc.cm, num_d
      {name: "clustering_regular", length_cut: 60 * wc.cm, flag_enable_extend: false},
      {name: "clustering_regular", length_cut: 30 * wc.cm, flag_enable_extend: true},
      {name: "clustering_parallel_prolong", length_cut: 35 * wc.cm},
      {name: "clustering_close", length_cut: 1.2 * wc.cm},
      {name: "clustering_extend_loop", num_try: 3},
      {name: "clustering_separate", use_ctpc: true},
      {name: "clustering_connect1"},
      {name: "clustering_degghost"},
      {name: "clustering_examine_x_boundary"},
      // {name: "clustering_protect_overclustering"},
      // {name: "clustering_neutrino"},
      {name: "clustering_isolated"},
    ],
  },
  nin=1, nout=1, uses=[geom_helper]},
```

```
118 local mabc = g.pnode({
119   type: "MultiAlgBlobClustering",
120   name: "%s-%d"%[anode.name, face],
121   data: {
122     inpath: "pointtrees/%d",
123     outpath: "pointtrees/%d",
124     perf: true,
125     bee_dir: bee_dir, // "data/0/0", // not used
126     bee_zip: "mabc-%s-face%d.zip"%[anode.name, face],
127     bee_detector: "sbnd",
128     initial_index: index, // New RSE configuration
129     use_config_rse: true, // Enable use of configured RSE
130     runNo: LrunNo,
131     subRunNo: LsubRunNo,
132     eventNo: LeventNo,
133     save_deadarea: true,
134     anode: wc.tn(anode),
135     face: face,
136     geom_helper: wc.tn(geom_helper),
137     func_cfgs: [
138       {name: "clustering_live_dead", dead_live_overlap_offset: 2},
139       {name: "clustering_extend", flag: 4, length_cut: 60 * wc.cm, num_try: 0
140       {name: "clustering_regular", length_cut: 60 * wc.cm, flag_enable_extend:
141       {name: "clustering_regular", length_cut: 30 * wc.cm, flag_enable_extend:
142       {name: "clustering_parallel_prolong", length_cut: 35 * wc.cm},
143       {name: "clustering_close", length_cut: 1.2 * wc.cm},
144       {name: "clustering_extend_loop", num_try: 3},
145       {name: "clustering_separate", use_ctpc: true},
146       {name: "clustering_connect1"},
147       {name: "clustering_degghost"},
148       {name: "clustering_examine_x_boundary"},
149       {name: "clustering_protect_overclustering"},
150       {name: "clustering_neutrino"},
151       {name: "clustering_isolated"},
152     ],
153   },
154   nin=1, nout=1, uses=[geom_helper]},
```

to-do 2

- ✓ merge previous matching dev with latest clustering
- ❑ cleanup and make multiple PRs

clus

ncluster check

bv: <https://www.phy.bnl.gov/twister/bee/set/383324da-5ecb-47b9-88b4-ce32c73c822c/event/0/>

hy: <https://www.phy.bnl.gov/twister/bee/set/4847e035-77cf-4d6e-a49b-f041d73553e3/event/0/>

```
grep "nclusters=" hy/wct.log
00:49:48.691] D [ clus ] <MultiAlgBlobClustering:> pre clustering grouping <Grouping [0x24277df0]: nclusters=329>
00:49:49.987] D [ clus ] <MultiAlgBlobClustering:> clustering_live_dead grouping <Grouping [0x24277df0]: nclusters=313>
00:49:50.151] D [ clus ] <MultiAlgBlobClustering:> clustering_extend grouping <Grouping [0x24277df0]: nclusters=310>
00:49:50.359] D [ clus ] <MultiAlgBlobClustering:> clustering_regular grouping <Grouping [0x24277df0]: nclusters=305>
00:49:50.508] D [ clus ] <MultiAlgBlobClustering:> clustering_regular grouping <Grouping [0x24277df0]: nclusters=304>
00:49:50.940] D [ clus ] <MultiAlgBlobClustering:> clustering_parallel_prolong grouping <Grouping [0x24277df0]: nclusters=303>
00:49:51.579] D [ clus ] <MultiAlgBlobClustering:> clustering_close grouping <Grouping [0x24277df0]: nclusters=248>
00:49:52.630] D [ clus ] <MultiAlgBlobClustering:> clustering_extend_loop grouping <Grouping [0x24277df0]: nclusters=236>
00:49:53.368] D [ clus ] <MultiAlgBlobClustering:> clustering_separate grouping <Grouping [0x24277df0]: nclusters=264>
00:49:53.963] D [ clus ] <MultiAlgBlobClustering:> clustering_connect1 grouping <Grouping [0x24277df0]: nclusters=194>
00:49:55.686] D [ clus ] <MultiAlgBlobClustering:> clustering_degghost grouping <Grouping [0x24277df0]: nclusters=105>
00:49:55.709] D [ clus ] <MultiAlgBlobClustering:> clustering_examine_x_boundary grouping <Grouping [0x24277df0]: nclusters=105>
00:49:57.248] D [ clus ] <MultiAlgBlobClustering:> clustering_protect_overclustering grouping <Grouping [0x24277df0]: nclusters=109>
00:49:57.420] D [ clus ] <MultiAlgBlobClustering:> clustering_isolated grouping <Grouping [0x24277df0]: nclusters=39>

ring grouping <Grouping [0x58f9470]: nclusters=329>
_live_dead grouping <Grouping [0x58f9470]: nclusters=313>
_extend grouping <Grouping [0x58f9470]: nclusters=310>
_regular grouping <Grouping [0x58f9470]: nclusters=305>
_regular grouping <Grouping [0x58f9470]: nclusters=304>
_parallel_prolong grouping <Grouping [0x58f9470]: nclusters=303>
_close grouping <Grouping [0x58f9470]: nclusters=248>
_extend_loop grouping <Grouping [0x58f9470]: nclusters=236>
_separate grouping <Grouping [0x58f9470]: nclusters=264>
_connect1 grouping <Grouping [0x58f9470]: nclusters=194>
_degghost grouping <Grouping [0x58f9470]: nclusters=105>
_examine_x_boundary grouping <Grouping [0x58f9470]: nclusters=105>
_protect_overclustering grouping <Grouping [0x58f9470]: nclusters=108>
_isolated grouping <Grouping [0x58f9470]: nclusters=39>
```

likely due to randomness

timing check

```
151 [00:49:58.178] D [ clus ] <MultiAlgBlobClustering:> MultiAlgBlobClustering performance summary:
```

```
152 Time summary:
```

```
153 TICK: 0 ms (this: 0 ms) starting MultiAlgBlobClustering
154 TICK: 296 ms (this: 296 ms) loaded live clusters
155 TICK: 339 ms (this: 42 ms) loaded dead clusters
156 TICK: 719 ms (this: 380 ms) loaded dump live clusters to bee
157 TICK: 729 ms (this: 10 ms) loaded dump dead regions to bee
158 TICK: 729 ms (this: 0 ms) pre clustering
159 TICK: 2025 ms (this: 1296 ms) clustering_live_dead
160 TICK: 2189 ms (this: 163 ms) clustering_extend
161 TICK: 2397 ms (this: 208 ms) clustering_regular
162 TICK: 2546 ms (this: 148 ms) clustering_regular
163 TICK: 2978 ms (this: 432 ms) clustering_parallel_prolong
164 TICK: 3617 ms (this: 638 ms) clustering_close
165 TICK: 4668 ms (this: 1051 ms) clustering_extend_loop
166 TICK: 5406 ms (this: 737 ms) clustering_separate
167 TICK: 6001 ms (this: 594 ms) clustering_connect1
168 TICK: 7724 ms (this: 1722 ms) clustering_degghost
169 TICK: 7747 ms (this: 23 ms) clustering_examine_x_boundary
170 TICK: 9286 ms (this: 1538 ms) clustering_protect_overclustering
171 TICK: 9458 ms (this: 172 ms) clustering_isolated
172 TICK: 9826 ms (this: 367 ms) dump live clusters to bee
173 TICK: 10031 ms (this: 205 ms) output live clusters to tensors
174 TICK: 10064 ms (this: 32 ms) output dead clusters to tensors
175 TICK: 10064 ms (this: 0 ms) combine tensors
176 TICK: 10207 ms (this: 142 ms) clear pc tree memory
177
```

```
172 [00:52:50.059] D [ clus ] <MultiAlgBlobClustering:> MultiAlgBl
```

```
173 Time summary:
```

```
174 TICK: 0 ms (this: 0 ms) starting MultiAlgBlobClustering
175 TICK: 301 ms (this: 301 ms) loaded live clusters
176 TICK: 344 ms (this: 43 ms) loaded dead clusters
177 TICK: 726 ms (this: 382 ms) loaded dump live clusters to bee
178 TICK: 737 ms (this: 10 ms) loaded dump dead regions to bee
179 TICK: 737 ms (this: 0 ms) pre clustering
180 TICK: 1999 ms (this: 1262 ms) clustering_live_dead
181 TICK: 2156 ms (this: 157 ms) clustering_extend
182 TICK: 2361 ms (this: 205 ms) clustering_regular
183 TICK: 2514 ms (this: 152 ms) clustering_regular
184 TICK: 2957 ms (this: 443 ms) clustering_parallel_prolong
185 TICK: 3575 ms (this: 617 ms) clustering_close
186 TICK: 4630 ms (this: 1055 ms) clustering_extend_loop
187 TICK: 5353 ms (this: 722 ms) clustering_separate
188 TICK: 5954 ms (this: 601 ms) clustering_connect1
189 TICK: 7637 ms (this: 1682 ms) clustering_degghost
190 TICK: 7661 ms (this: 24 ms) clustering_examine_x_boundary
191 TICK: 9147 ms (this: 1485 ms) clustering_protect_overclustering
192 TICK: 9323 ms (this: 175 ms) clustering_isolated
193 TICK: 9693 ms (this: 370 ms) dump live clusters to bee
194 TICK: 9900 ms (this: 206 ms) output live clusters to tensors
195 TICK: 9933 ms (this: 33 ms) output dead clusters to tensors
196 TICK: 9933 ms (this: 0 ms) combine tensors
197 TICK: 10077 ms (this: 144 ms) clear pc tree memory
198
```

merging two branches

merged back to apply-pointcloud

97c212cafc8fe29a32ac8bf69e665aaeebf8da6a

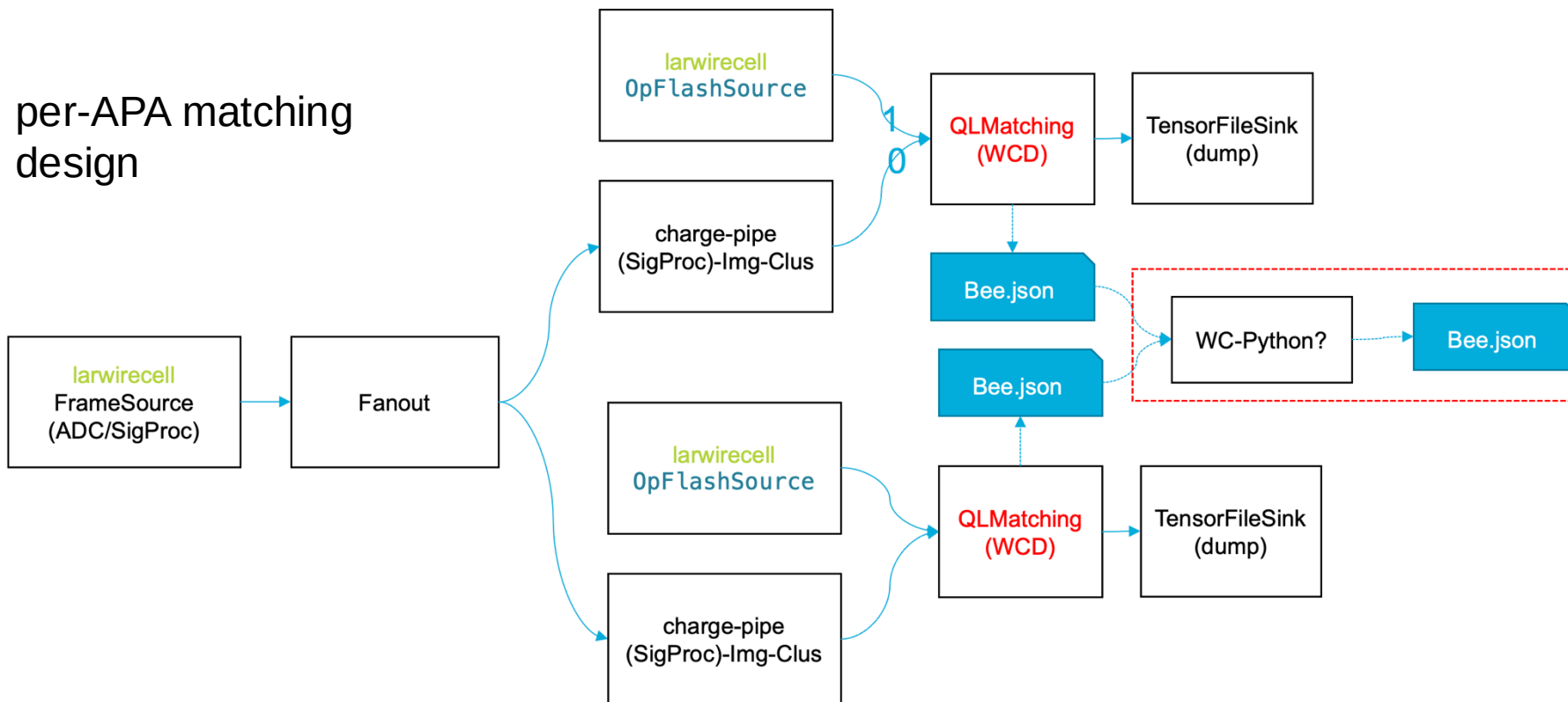
<https://www.phy.bnl.gov/twister/bee/set/ce807e5c-bd6d-4bd2-b970-f1e7e6830f81/event/0/>

```
$grep ": nclusters=" wct.log
[01:12:34.394] D [ clus ] <MultiAlgBlobClustering:> pre clustering grouping <Grouping [0xaac61d0]: nclusters=329>
[01:12:35.667] D [ clus ] <MultiAlgBlobClustering:> clustering_live_dead grouping <Grouping [0xaac61d0]: nclusters=313>
[01:12:35.821] D [ clus ] <MultiAlgBlobClustering:> clustering_extend grouping <Grouping [0xaac61d0]: nclusters=310>
[01:12:36.023] D [ clus ] <MultiAlgBlobClustering:> clustering_regular grouping <Grouping [0xaac61d0]: nclusters=305>
[01:12:36.194] D [ clus ] <MultiAlgBlobClustering:> clustering_regular grouping <Grouping [0xaac61d0]: nclusters=304>
[01:12:36.627] D [ clus ] <MultiAlgBlobClustering:> clustering_parallel_prolong grouping <Grouping [0xaac61d0]: nclusters=303>
[01:12:37.235] D [ clus ] <MultiAlgBlobClustering:> clustering_close grouping <Grouping [0xaac61d0]: nclusters=248>
[01:12:38.290] D [ clus ] <MultiAlgBlobClustering:> clustering_extend_loop grouping <Grouping [0xaac61d0]: nclusters=236>
[01:12:39.037] D [ clus ] <MultiAlgBlobClustering:> clustering_separate grouping <Grouping [0xaac61d0]: nclusters=264>
[01:12:39.621] D [ clus ] <MultiAlgBlobClustering:> clustering_connect1 grouping <Grouping [0xaac61d0]: nclusters=196>
[01:12:41.339] D [ clus ] <MultiAlgBlobClustering:> clustering_degghost grouping <Grouping [0xaac61d0]: nclusters=105>
[01:12:41.362] D [ clus ] <MultiAlgBlobClustering:> clustering_examine_x_boundary grouping <Grouping [0xaac61d0]: nclusters=105>
[01:12:42.878] D [ clus ] <MultiAlgBlobClustering:> clustering_protect_overclustering grouping <Grouping [0xaac61d0]: nclusters=108>
[01:12:43.053] D [ clus ] <MultiAlgBlobClustering:> clustering_isolated grouping <Grouping [0xaac61d0]: nclusters=39>
```

matching

Charge-light matching for SBND

- an actual cfg: <https://www.phy.bnl.gov/~hyu/sbnd/op/wcls-matching.pdf>



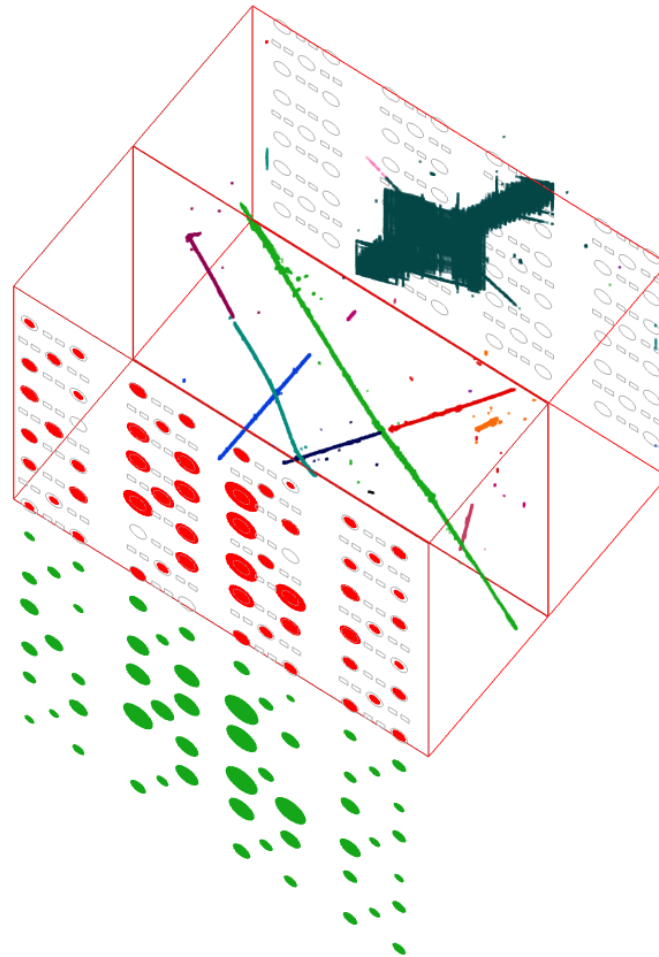
updated larwirecell+cfg to run latest clustering with matching

tes0 (previous placeholder clus): <https://www.phy.bnl.gov/twister/bee/set/d9c82567-2673-4b3a-9362-2eb3160f7f63>

test1 (data) (us): <https://www.phy.bnl.gov/twister/bee/set/a19ee912-db5b-4e01-b806-86ea1d2fa5c0/event/279/>

test1 (data) (ns): <https://www.phy.bnl.gov/twister/bee/set/e8bf4540-1644-4210-a8fe-50f450460ade/event/279/>

question: unit of flash time?



plan

Thoughts

Key issue:

- SBND is moving fast, we may lose our community if the tools are not available soon

Options:

1. Keep “clus” style translation with PCTree as the basis and try to do it as fast as we can
2. Update WCP to take WCT input and make it configurable a combined SBND detector

Intended timeline

2024/S4	2025/S1	2025/S2	2025/S3	2025/S4
Clustering – dev.	Clustering– dep. QL-Matching – dev. Traj. fitting – dev. muon tag – dev.	muon tag – dep. 3D PatRec – dev.	3D PatRec - dev AI/ML on vertexing/nu tagging/clustering/EE	3D PatRec - dep AI/ML on vertexing/nu tagging/clustering/EE

dev: development; dep: deployment

Pro/Con

Option 1: PCTree based PatRec in WCT:

- Pro:
 - easier to make things configurable
 - cleaner dependency
 - better long-term maintainability
- Con:
 - Need time to adapt things – may take longer

Option 2: Improve WCP to make it **configurable** and **accepting WCT::ITensorSet**

- Pro:
 - Not TOO ugly:
 - Essentially making a WCT plugin that **depends on WCP**, providing a **uboonePatRec ITensorSetFilter**
 - **Should be faster (but not free still work to do)**
 - We gain experience doing this work
 - Benefit the final version
 - Potentially less bugs from 1st round
- Con:
 - Usage potentially limited
 - May need a little bit duplicated work for the final version for DUNE

reco1.5:

uboone_nusel_app/apps/prod-wire-cell-matching-nusel.cxx

- [WCP2dToy::Clustering_jump_gap_cosmics](#)
- WCP2dToy::tpc_light_match
- WCP2dToy::ExamineBundles
- IO

larwirecell: ITen → ITen

reco2:

pid/apps/wire-cell-prod-stm.cxx

- WCPPID/ToyFiducial
- WCPPID::PR3DCluster
- standalone?

WCP/tio: ITen → WCP Objects → ITen

pid/apps/wire-cell-prod-nue.cxx

- WCPPID::calc_sampling_points
- WCPPID::Protect_Over_Clustering
- WCPPID::NeutrinoID

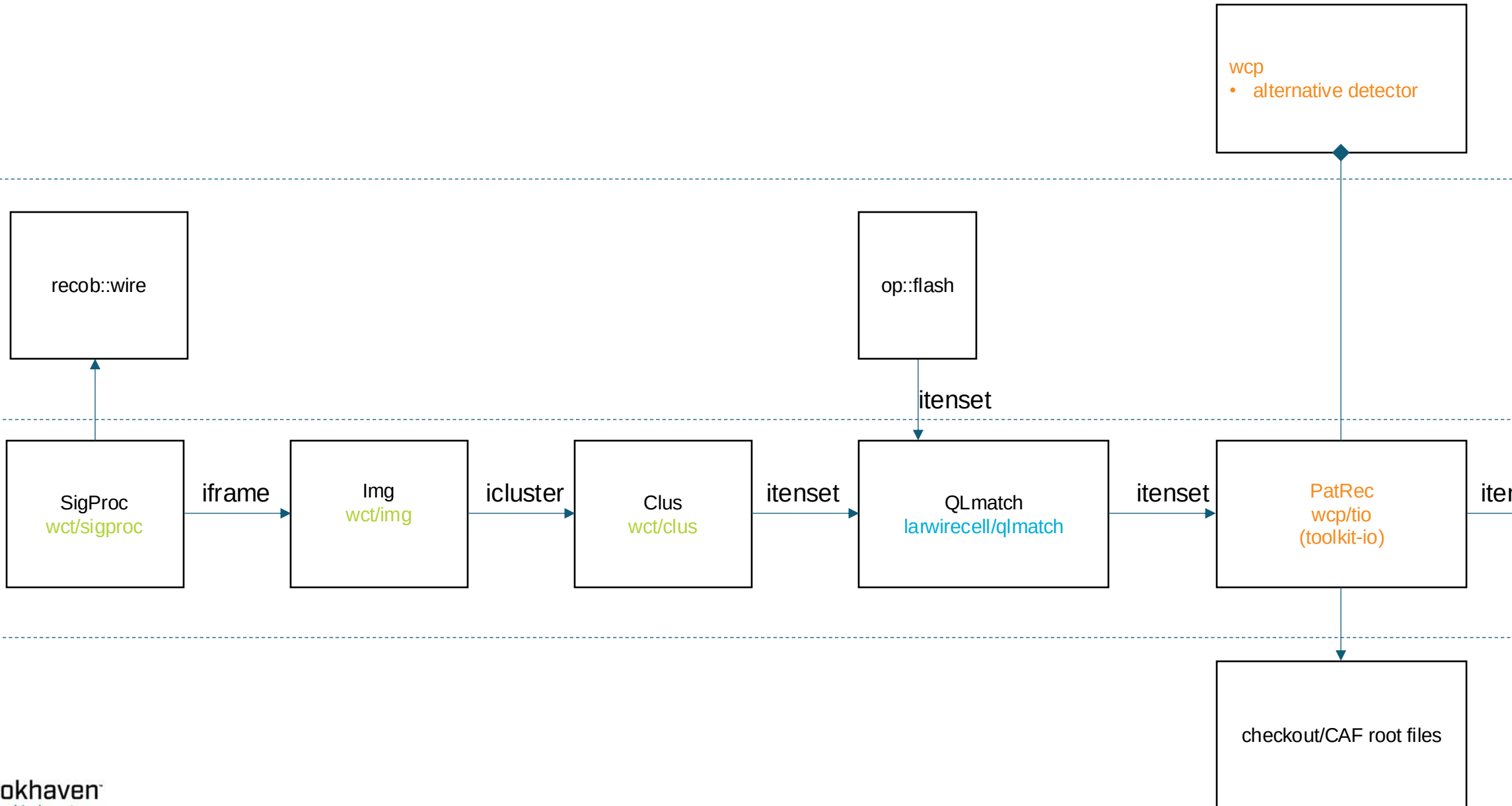
Option 2: flowchart

Wire-Cell
Prototype

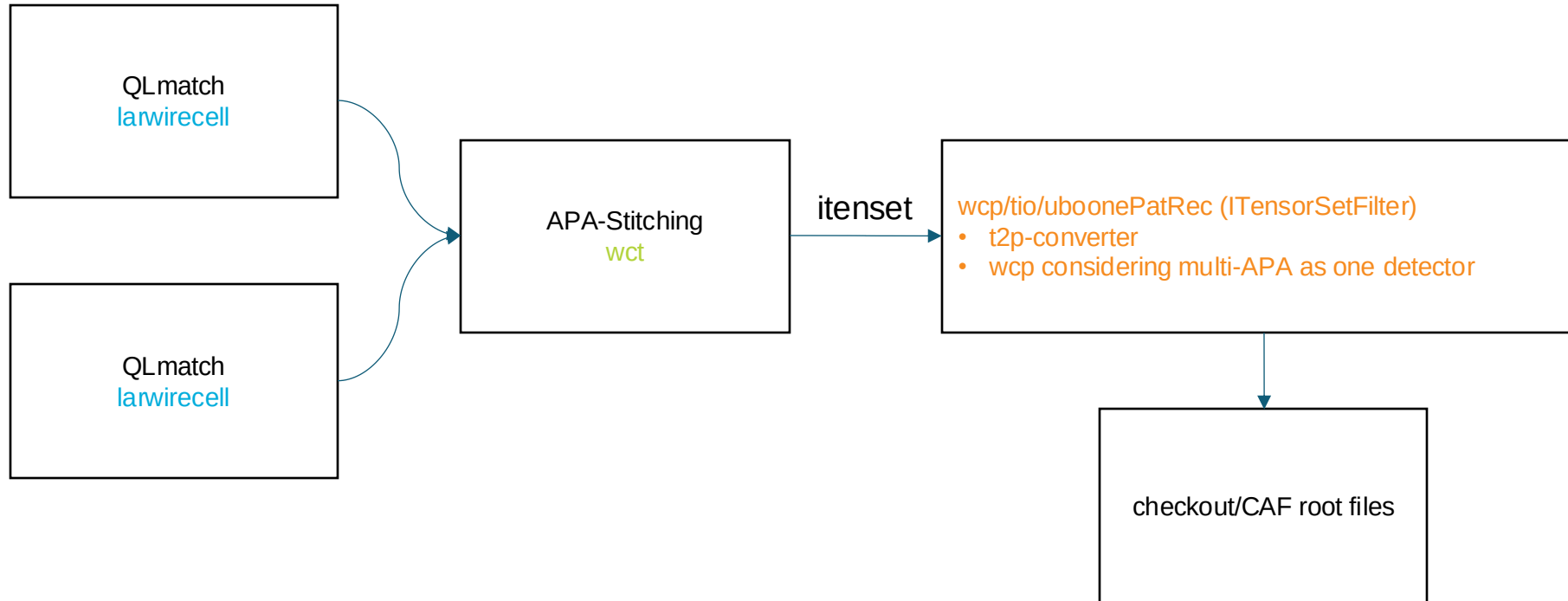
artROOT

Wire-Cell
Toolkit

Standalone



Option 2: flowchart



Comments from Tuesday discussion

- Xin: Direct uboone code may won't have the best performance on SBND
 - Need trained people to understand the algorithm details and be able to adjust for SBND
 - one of the main motivation of our porting campain
- Xin: Pure implementation time could be not much different for Option 1 and 2. Xin and Brett should have a way to quickly implement option 1 with certain compromises in coding styling.
- Haiwang, Hanyu and UChicago should try finish QL-matching within ~1month

backups