Hashing Classification for charged particle tracking

Luiza Adelina Ciucu (ATLAS)

19 June 2020

UNIVERSITÉ
DE GENÈVE

FACULTY OF SCIENCE
Physics Section

Introduction

- Compare 3 methods to balance nb of positive and negative hits:
  - A: Unbalanced 30 events. Train 2.29M buckets. Test 0.95 buckets.
  - B: Balanced 30 events. Train 1.39 M buckets. Test 0.61 buckets.
  - Not trained: Unbalanced 100 events. Train 7.35 M buckets. Test 3.22.
  - C: Balanced2 100 events. Train 2.21M buckets. Test 0.98 buckets.
- A and C similar nb of buckets, so a fair comparison.
- For each group of 10: 7 in train, 3 in test.
- A=Unbalanced: keep all buckets.
- B=Balanced: remove buckets (from the left) so that bucket distribution is symmetric around 10 in NbPositiveHit per bucket.
- C=Balanced2: similar to B, but cut the peak so that same number of buckets between 6-14 inclusive.
  Also 0,1, 19, 20 are kept as they are.
  But 2, 3, 4, 5, 15, 16, 17, 18 remain as in B.
- Used min 0, 4, 7 and 10 positive hits in the bucket; if less, consider all hits in the bucket to be negative, but use same balancing as above.

Introduction 2

- Reminder how a bucket is created:
  - Loop over events, and for each event build annoy index and then:
  - Loop over hits and for each hit build a bucket using annoy and 20 nearest neighbours by direction
  - Loop over hits in the bucket, find their particleID; find particleID with most hits in the bucket; denote it majority particle.
  - Loop over hits in the bucket again, if belongs to the majority particle assign output $+1$, else -1.

Bucket balancing procedure 1/4

- Remove buckets so that bucket distribution is symmetric around 10 in NbPositiveHit per bucket (with default weight of 1.0): Train and Test.
- Input: `nparray_output` (Nx20)
- Output: `nparray_outputBalanced` (Nx20)
- Step 1: count buckets for each category of nbPositiveHit
  - output: `dict_nbPositiveHit_counterBucket`
  - process: loop over buckets, for each bucket count positive hits into nbPositiveHit, increase counter in dicti at that key of nbPositiveHit.
- The result is shown below for Train. Note not balanced around 10.

```
Original unbalanced. Train:
nbPositiveHit=0 counterBucket=0 percentBucket=0.0
nbPositiveHit=1 counterBucket=0 percentBucket=0.0
nbPositiveHit=2 counterBucket=3973 percentBucket=0.2
nbPositiveHit=3 counterBucket=33349 percentBucket=1.5
nbPositiveHit=4 counterBucket=115266 percentBucket=5.0
nbPositiveHit=5 counterBucket=186516 percentBucket=8.1
nbPositiveHit=6 counterBucket=270380 percentBucket=11.8
nbPositiveHit=7 counterBucket=294847 percentBucket=12.9
nbPositiveHit=8 counterBucket=298765 percentBucket=13.0
nbPositiveHit=9 counterBucket=260813 percentBucket=11.4
nbPositiveHit=10 counterBucket=262604 percentBucket=11.5
nbPositiveHit=11 counterBucket=194065 percentBucket=8.5
nbPositiveHit=12 counterBucket=157420 percentBucket=6.9
nbPositiveHit=13 counterBucket=99938 percentBucket=4.4
nbPositiveHit=14 counterBucket=66595 percentBucket=2.9
nbPositiveHit=15 counterBucket=29752 percentBucket=1.3
nbPositiveHit=16 counterBucket=10828 percentBucket=0.5
nbPositiveHit=17 counterBucket=3184 percentBucket=0.1
nbPositiveHit=18 counterBucket=1560 percentBucket=0.1
nbPositiveHit=19 counterBucket=379 percentBucket=0.0
nbPositiveHit=20 counterBucket=148 percentBucket=0.0
```

Bucket balancing procedure 2/4

- Step 2: from this (unbalanced) dict calculate desired balanced dict.
  - input: `dict_nbPositiveHit_counterBucket`
  - output: `dict_nbPositiveHit_counterBucket_Balanced`
  - process: loop over i from the first half of nbPositiveHit (from 0 to 10)
  - nbLeft=`dict_nbPositiveHit_counterBucket[i]`
  - nbRight=`dict_nbPositiveHit_counterBucket[20-i]`
  - Find nbMin from nbLeft and nbRight
  - Set in the new balanced dictionary both values ot the nbMin
  - `dict_nbPositiveHit_counterBucke_Balancedt[i]=nbMin`
  - `dict_nbPositiveHit_counterBucke_Balancedt[20-i]=nbMin`
- The result is shown below for Train. Note it is balanced around 10.
- Right usually smaller, so remains the same and remove from left.
- nbPositiveHit 0 and 1 have counts of 0, $\rightarrow$ set 19 and 20 to zero.

```
Original unbalanced. Train:
nbPositiveHit=0 counterBucket=0 percentBucket=0.0
nbPositiveHit=1 counterBucket=0 percentBucket=0.0
nbPositiveHit=2 counterBucket=3973 percentBucket=0.2
nbPositiveHit=3 counterBucket=33349 percentBucket=1.5
nbPositiveHit=4 counterBucket=115266 percentBucket=5.0
nbPositiveHit=5 counterBucket=186516 percentBucket=8.1
nbPositiveHit=6 counterBucket=270380 percentBucket=11.8
nbPositiveHit=7 counterBucket=294847 percentBucket=12.9
nbPositiveHit=8 counterBucket=298765 percentBucket=13.0
nbPositiveHit=9 counterBucket=260813 percentBucket=11.4
nbPositiveHit=10 counterBucket=262604 percentBucket=11.5
nbPositiveHit=11 counterBucket=194065 percentBucket=8.5
nbPositiveHit=12 counterBucket=157420 percentBucket=6.9
nbPositiveHit=13 counterBucket=99938 percentBucket=4.4
nbPositiveHit=14 counterBucket=66595 percentBucket=2.9
nbPositiveHit=15 counterBucket=29752 percentBucket=1.3
nbPositiveHit=16 counterBucket=10828 percentBucket=0.5
nbPositiveHit=17 counterBucket=3184 percentBucket=0.1
nbPositiveHit=18 counterBucket=1560 percentBucket=0.1
nbPositiveHit=19 counterBucket=379 percentBucket=0.0
nbPositiveHit=20 counterBucket=148 percentBucket=0.0
```

Bucket balancing procedure 3/4

- Step 3: use desired balanced dict to obtain balanced output.
  - input: `dict_nbPositiveHit_counterBucket_Balanced`
  - input: `nparray_output` (Nx20)
  - output: `nparray_outputBalanced` (Nx20)
  - process: loop over buckets from `nparray_output`:
  - find nbPositiveHit for the bucket
  - from nbPositiveHit find desired number of bucket in this category
  - count the current number of buckets in this category
  - if current counter $\leq$ desired number, append to a list
  - else (do nothing, so skip it)
  - .
  - after for loop convert list to `nparray_outputBalanced`
  - as in step 1, calculate dictionary of counterBucket for each category
  - by printing verify it is the same as the one desired (confirmed)
  - next overlay histograms for nbPositiveHit in Unbalanced and Balanced
  - as expected, now it is balanced, the right side is the same in both cases, and we removed buckets from the left
  - But we set 19 and 20 to zero, to keep as 0 and 1. Though 0 will always have no buckets. But otherwise makes the symmetry harder.

Bucket balancing method 2

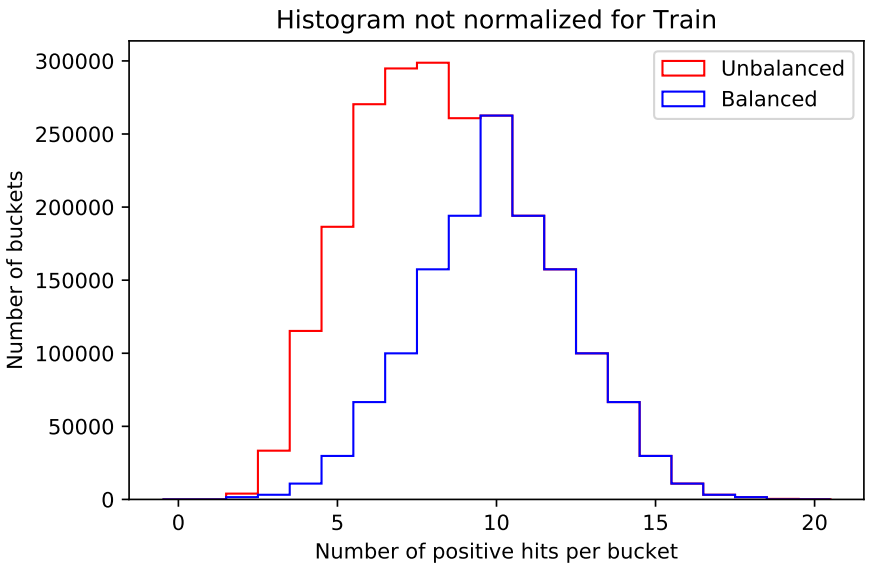
- Similar to Balanced from before (above), but with some changes.
- Cut the peak so that same number of buckets between 6-14 inclusive.
- Also 0,1, 19, 20 are kept as they are.
  - No buckets with nbPositiveHit=0, so it is unfair to set nbPositiveHit=20 to zero as well.
  - Only 3 nbPositiveHit=3, so it is unfair to set nbPositiveHit=19 to 3.
- But 2, 3, 4, 5, 15, 16, 17, 18 remain as in B.

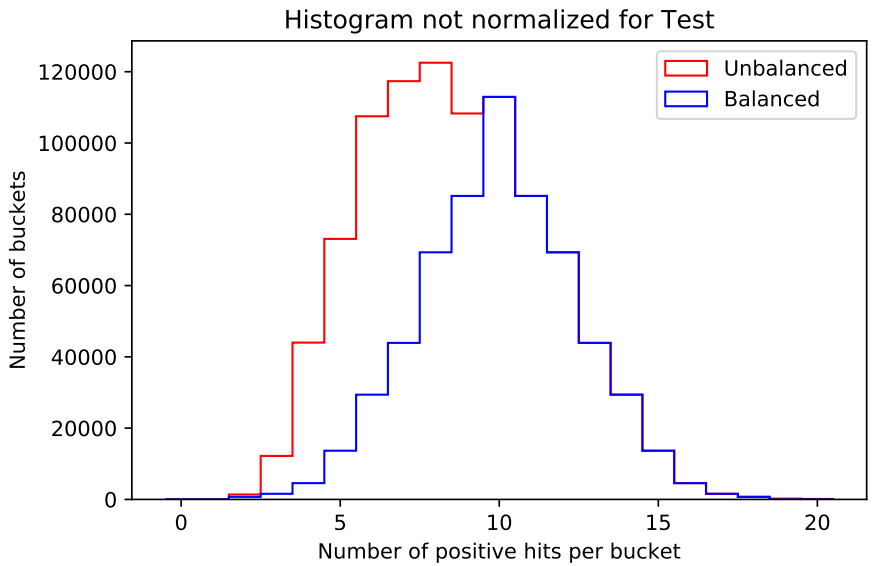
```
100 events, Trian, Unbalanced:

nbPositiveHit=00 counterBucket=0 percentBucket=0.0
nbPositiveHit=01 counterBucket=3 percentBucket=0.0
nbPositiveHit=02 counterBucket=11977 percentBucket=0.2
nbPositiveHit=03 counterBucket=102408 percentBucket=1.4
nbPositiveHit=04 counterBucket=363809 percentBucket=4.9
nbPositiveHit=05 counterBucket=591638 percentBucket=8.0
nbPositiveHit=06 counterBucket=860452 percentBucket=11.7
nbPositiveHit=07 counterBucket=943536 percentBucket=12.8
nbPositiveHit=08 counterBucket=966108 percentBucket=13.1
nbPositiveHit=09 counterBucket=840738 percentBucket=11.4
nbPositiveHit=10 counterBucket=847413 percentBucket=11.5
nbPositiveHit=11 counterBucket=627344 percentBucket=8.5
nbPositiveHit=12 counterBucket=512063 percentBucket=7.0
nbPositiveHit=13 counterBucket=324381 percentBucket=4.4
nbPositiveHit=14 counterBucket=213198 percentBucket=2.9
nbPositiveHit=15 counterBucket=95757 percentBucket=1.3
nbPositiveHit=16 counterBucket=34417 percentBucket=0.5
nbPositiveHit=17 counterBucket=10756 percentBucket=0.1
nbPositiveHit=18 counterBucket=5544 percentBucket=0.1
nbPositiveHit=19 counterBucket=1406 percentBucket=0.0
nbPositiveHit=20 counterBucket=594 percentBucket=0.0
```

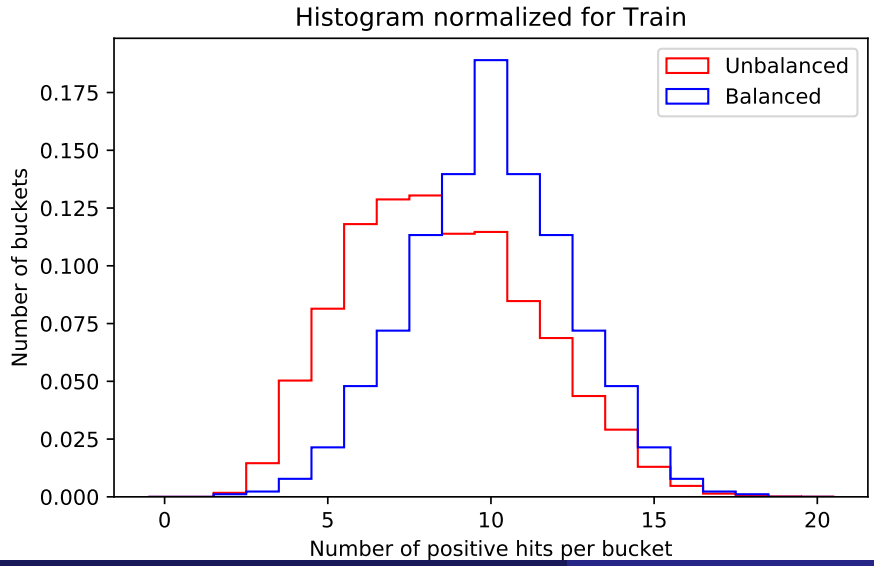
```
100 events, Train, Balanced2:

nbPositiveHit=00 counterBucket=0 percentBucket=0.0
nbPositiveHit=01 counterBucket=3 percentBucket=0.0
nbPositiveHit=02 counterBucket=5544 percentBucket=0.1
nbPositiveHit=03 counterBucket=10756 percentBucket=0.1
nbPositiveHit=04 counterBucket=34417 percentBucket=0.5
nbPositiveHit=05 counterBucket=95757 percentBucket=1.3
nbPositiveHit=06 counterBucket=213198 percentBucket=2.9
nbPositiveHit=07 counterBucket=213198 percentBucket=2.9
nbPositiveHit=08 counterBucket=213198 percentBucket=2.9
nbPositiveHit=09 counterBucket=213198 percentBucket=2.9
nbPositiveHit=10 counterBucket=213198 percentBucket=2.9
nbPositiveHit=11 counterBucket=213198 percentBucket=2.9
nbPositiveHit=12 counterBucket=213198 percentBucket=2.9
nbPositiveHit=13 counterBucket=213198 percentBucket=2.9
nbPositiveHit=14 counterBucket=213198 percentBucket=2.9
nbPositiveHit=15 counterBucket=95757 percentBucket=1.3
nbPositiveHit=16 counterBucket=34417 percentBucket=0.5
nbPositiveHit=17 counterBucket=10756 percentBucket=0.1
nbPositiveHit=18 counterBucket=5544 percentBucket=0.1
nbPositiveHit=19 counterBucket=1406 percentBucket=0.0
nbPositiveHit=20 counterBucket=594 percentBucket=0.0
```

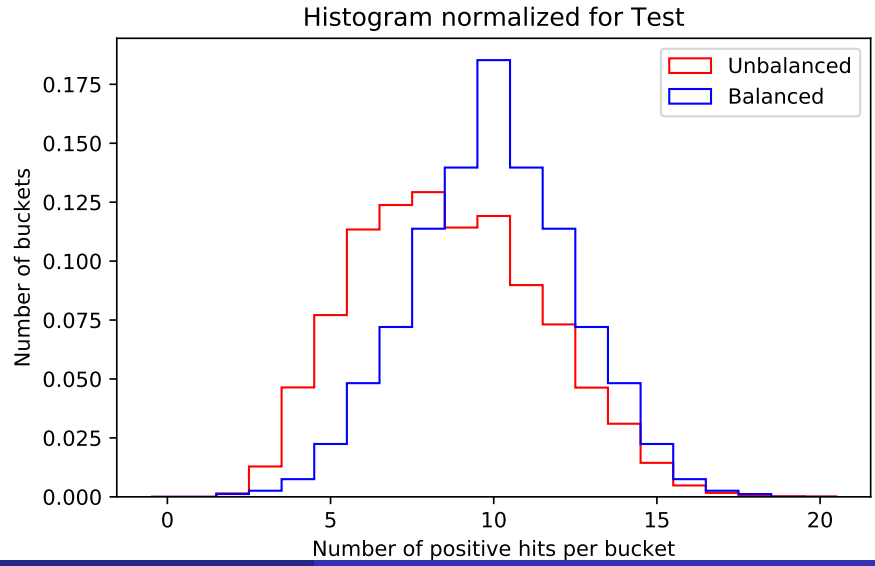
Histogram NbBuckets vs NbPositiveHit in a bucket.

- 30 events. Unbalanced (Balanced) are not (are) symmetric around 10.



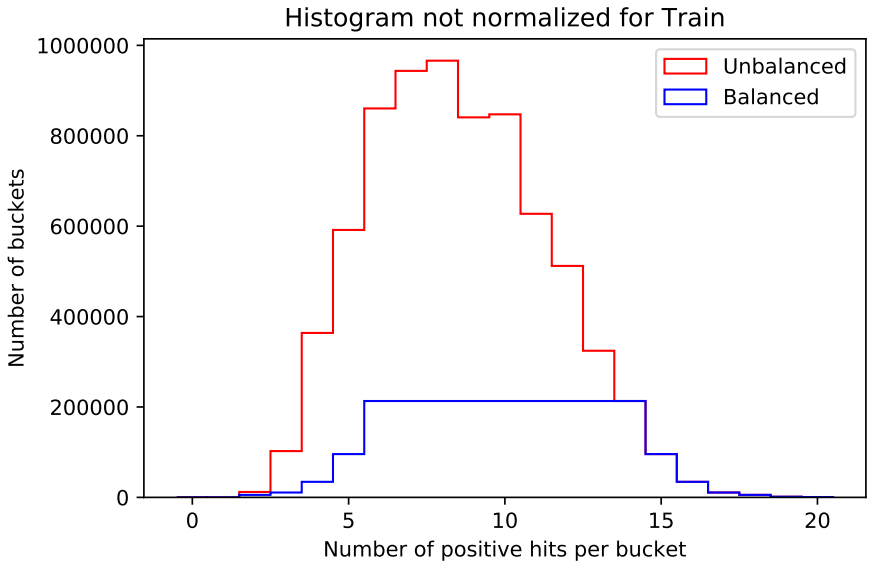


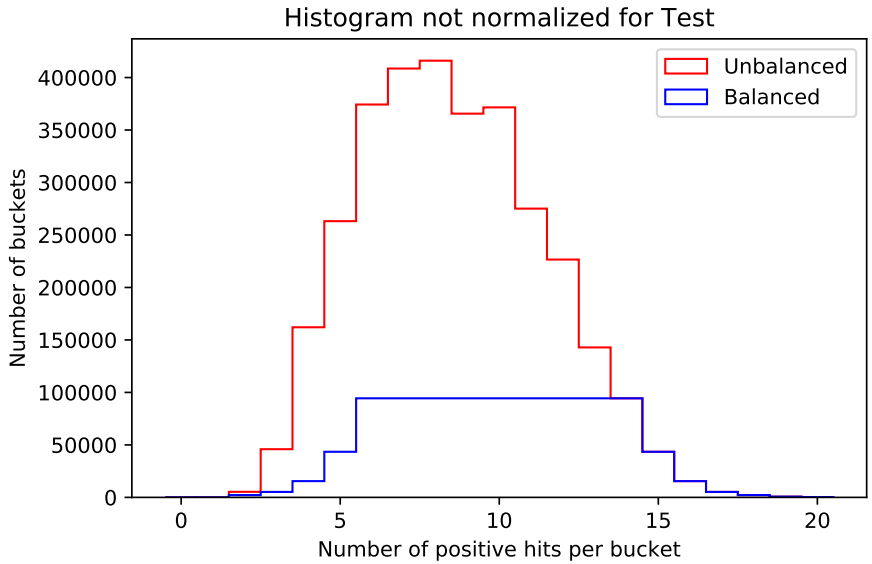


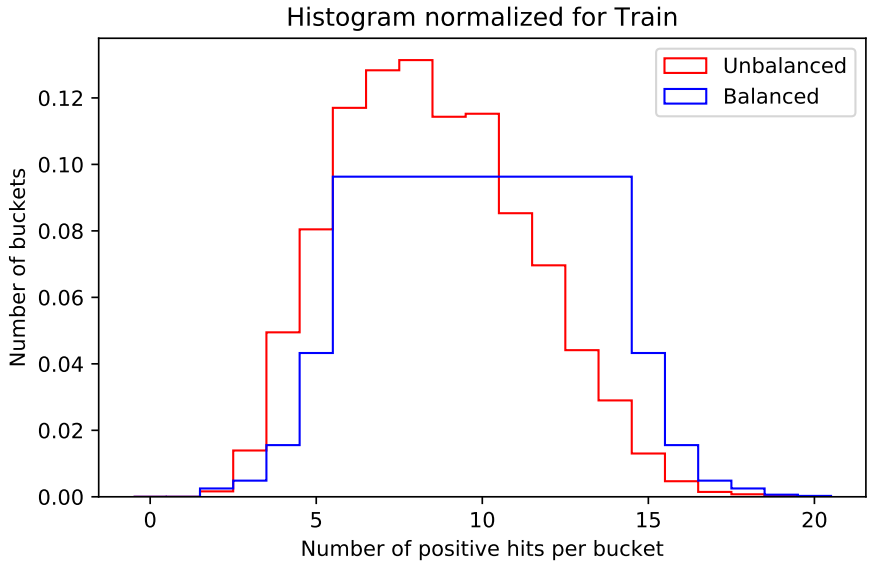


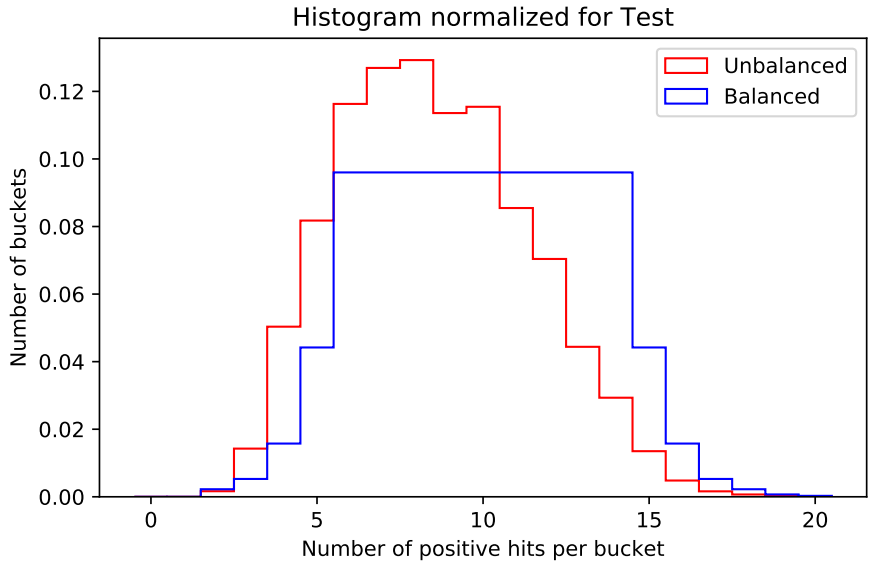
Histogram NbBuckets vs NbPositiveHit in a bucket.

- 100 events. Unbalanced vs Balanced2.



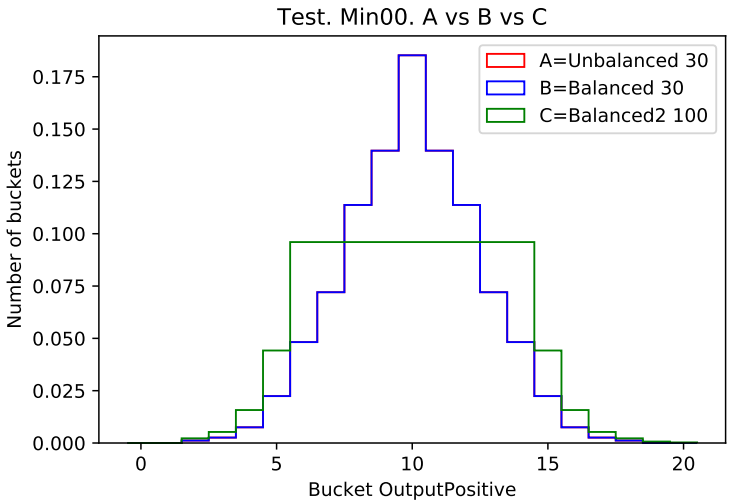


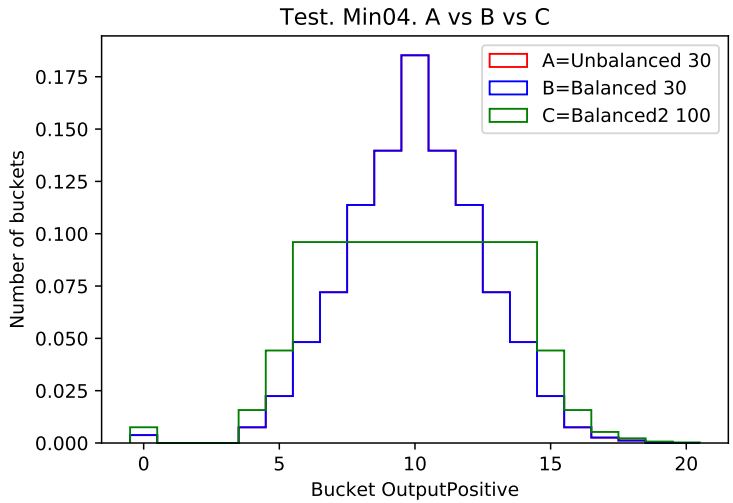


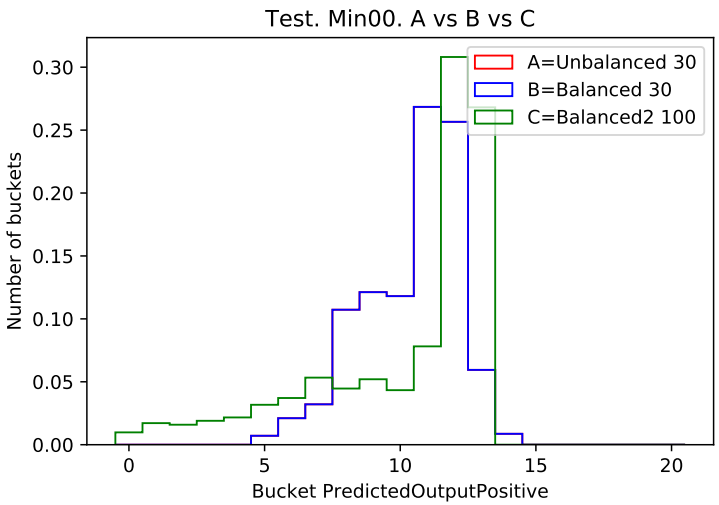


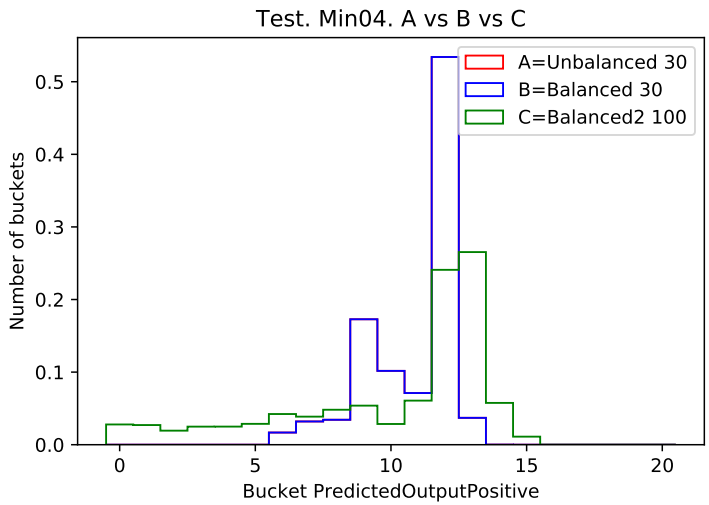
OutputPositive and PredictedOutputPositive 1/2

- Min00 (left) and Min04 (right)



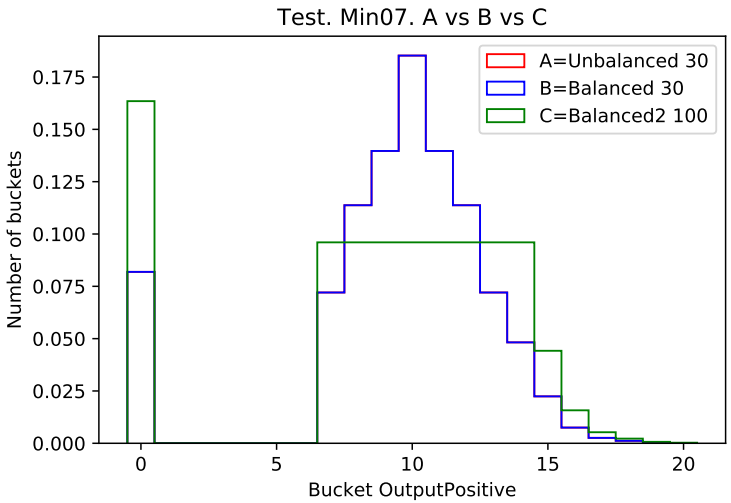


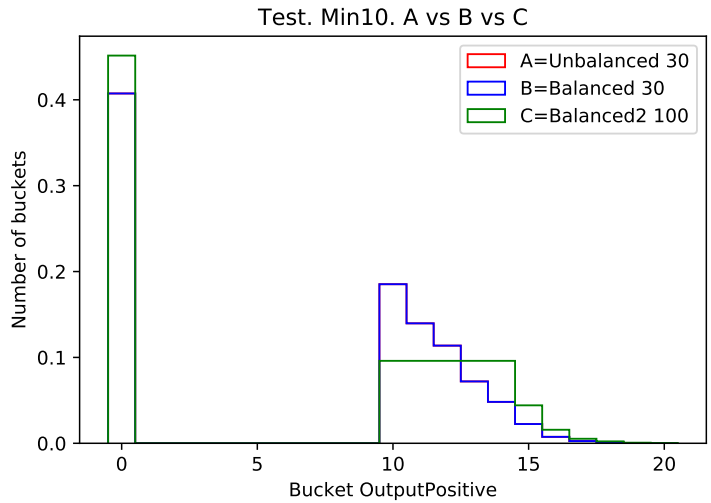


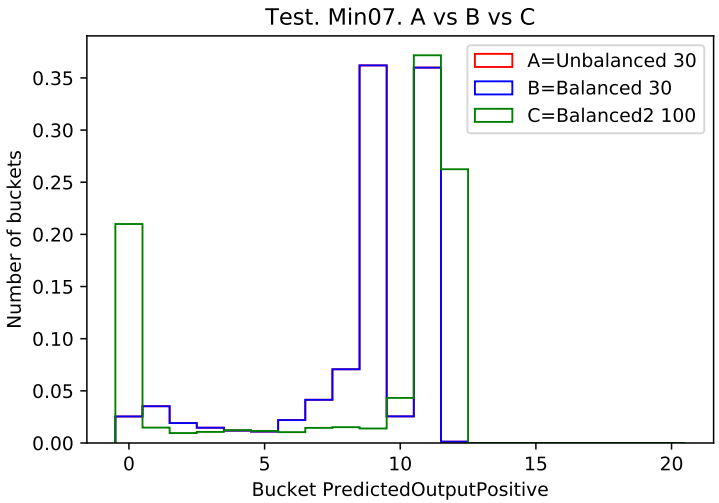


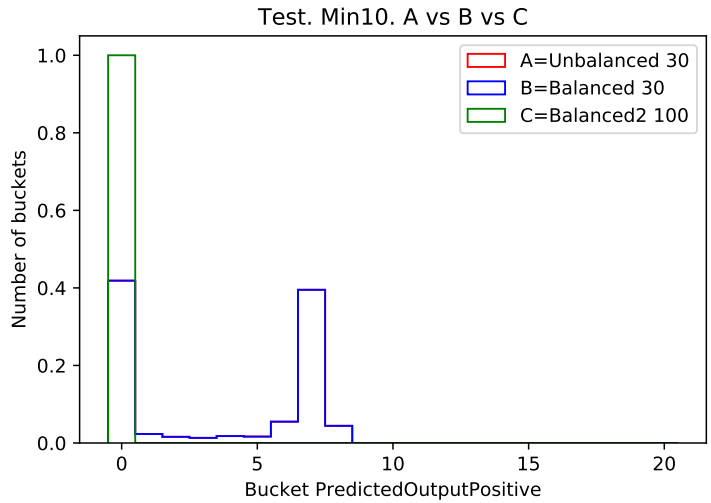
OutputPositive and PredictedOutputPositive 2/2

- Min07 (left) and Min10 (right).
- Min10 and method Unbalanced predicts all hits to be negative.





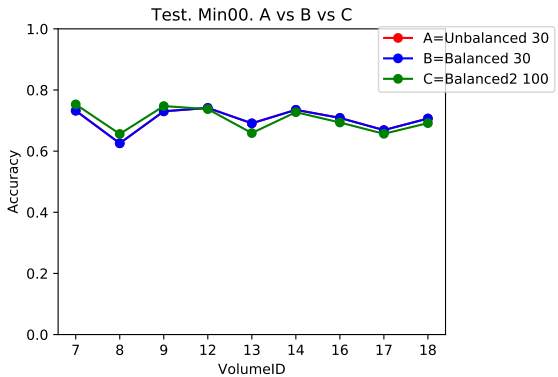


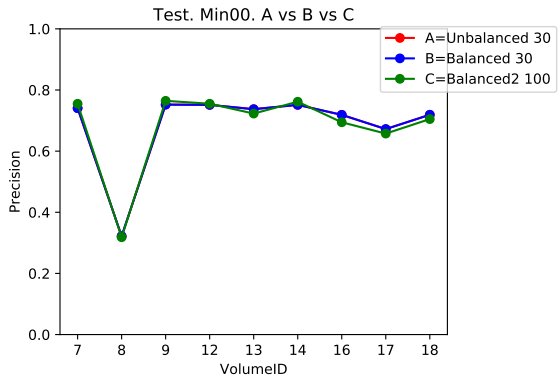


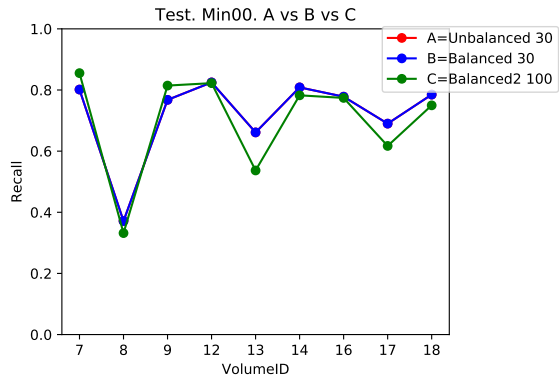
Metrics for each VolumeID overlay two methods 1/2.

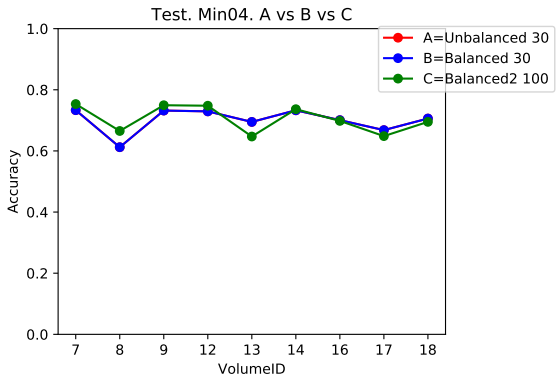
- Min00 and Min04 very similar.

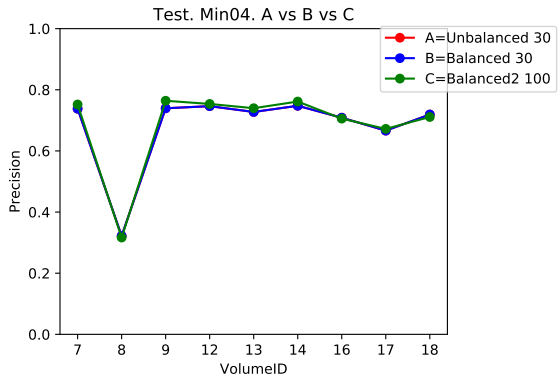
Accuracy	Precision	Recall
$\frac{TP+TN}{TP+FP+FN+TN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$

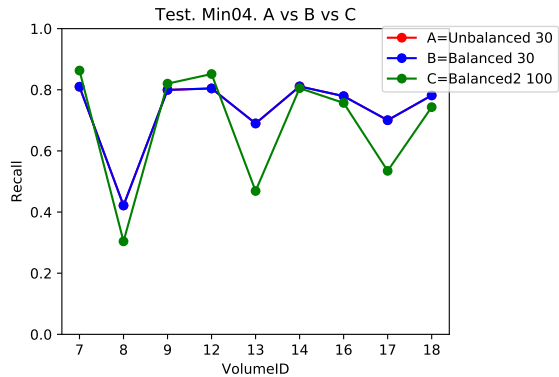








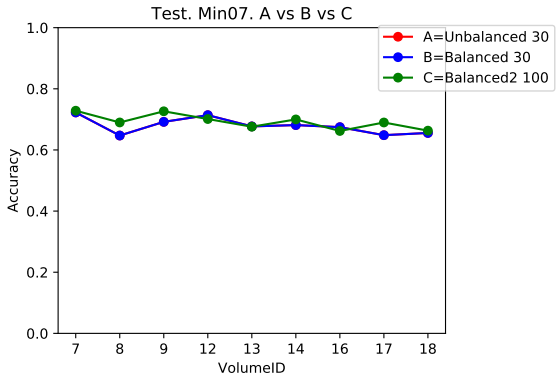


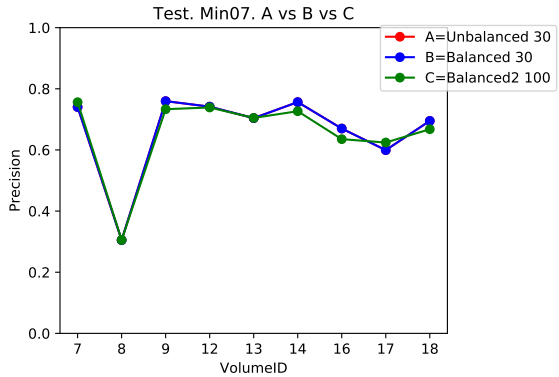


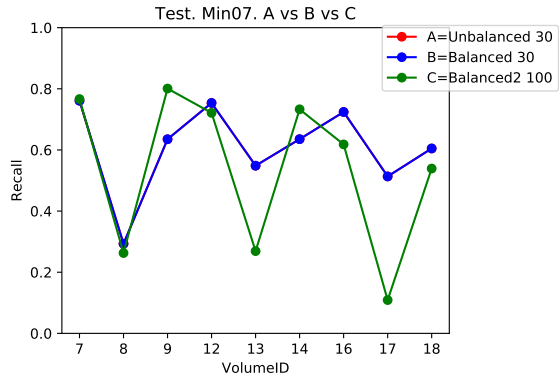
Metrics for each VolumeID overlay two methods 2/2.

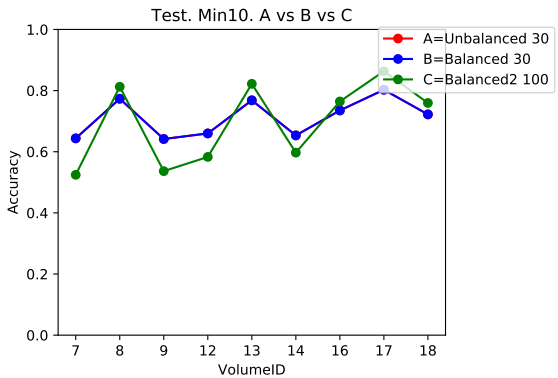
-
- Min10 learns all hits to be negative, so precision and recall at zero.

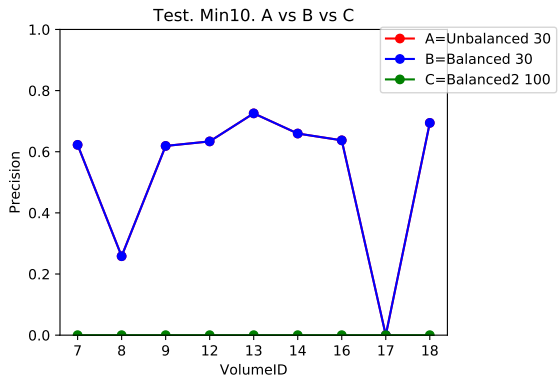
Accuracy	Precision	Recall
$\frac{TP+TN}{TP+FP+FN+TN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$

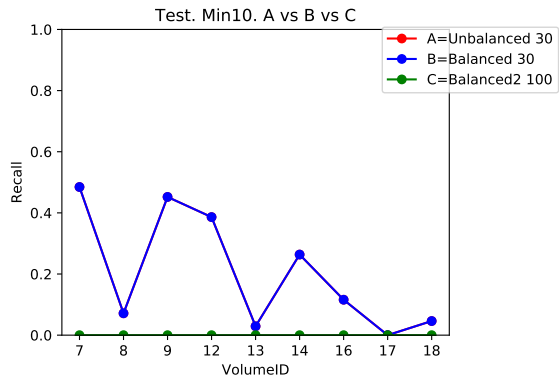






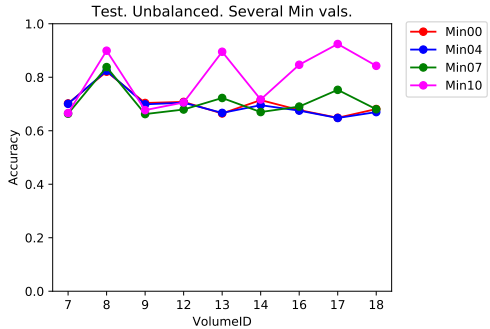


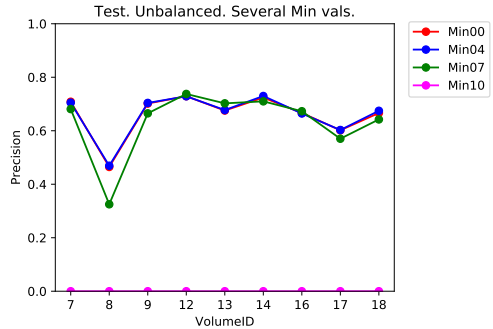


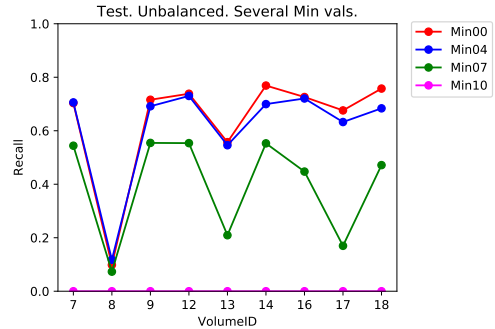


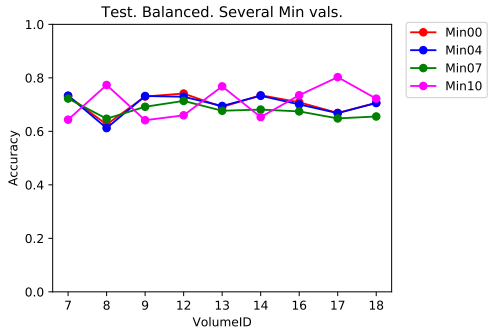
Metrics for each VolumeID with min value used.

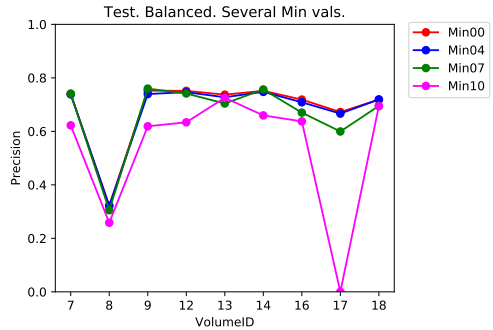
- Min00 and Min04 very similar.
- Min10 learns all hits to be negative, so precision and recall at zero.

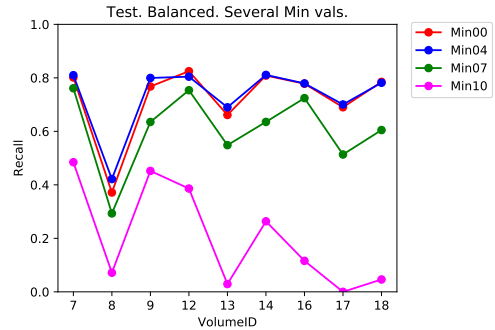


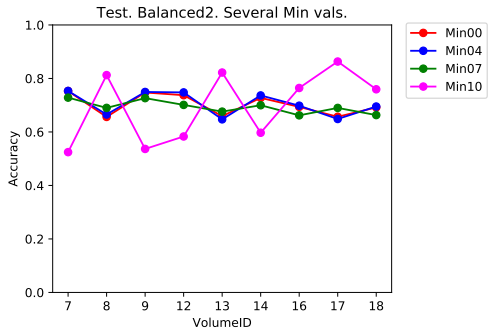


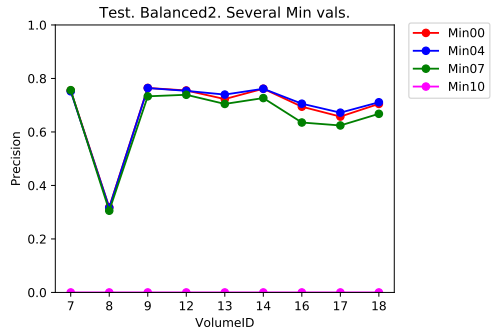


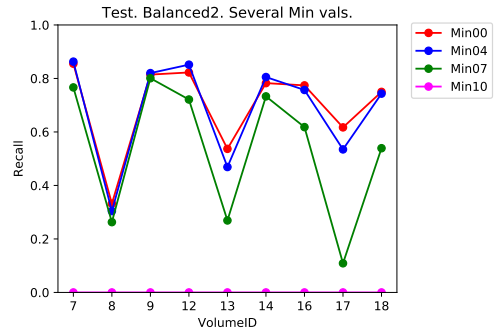












Conclusion. Future plans.

- Conclusions:
  - Compared 3 methods: Unbalanced vs Balanced vs Balanced2.
  - Balanced: remove buckets such that number of buckets with a given nbPositiveHit is symmetric around 10.
  - Balanced2: Same as Balanced, but reduce peak to have a flatter distribution (same values between 6-14), keep also 19 and 20 to non-zero.
  - Min00 and Min04 are very similar.
  - Min07 in between Min04 and Min10.
  - Overall, balancing buckets improve performance.
  - **Could choose 100 events, Balanced2, with Min04.**
- Future plans:
  - Balance buckets in the η of the majority-particle.
  - Write master thesis.