Adaptive Filters for Track Finding

A. Strandlie, University College Gjøvik, Norway
R. Frühwirth, HEPHY, Vienna, Austria

ACAT 2005, Zeuthen
May 22–27, 2005

Outline

- Introduction
- The combinatorial KF
- Four adaptive filters
- The simulation experiments
- Results
- Discussion and outlook

Introduction

- The Kalman filter (Billoir 1984, Frühwirth 1987) is a very popular method of track reconstruction.
- It is recursive: measurements are included one after the other.
- Alternation of propagation and update steps.
- Kalman filter can be used also for track finding (Billoir 1989).
- Current track candidate can be used in searching for compatible hits in the next detector layer.
- Selecting the most compatible hit is far from optimal in high track density environments.

Introduction

- The currently most powerful approach in this respect is the so-called combinatorial Kalman filter (Mankel 1997), which builds up a combinatorial tree of track candidates starting from a track seed.
- Recently, adaptive generalizations of the Kalman filter have been investigated.
- Suitable for solving the final assignment problem in parallel with the parameter estimation.
- Here, we present the adaptive filters as alternatives to the combinatorial Kalman filter (CKF) for the entire track finding procedure.

The combinatorial KF

- Start from "seed".
- Iterate the following steps:
  - Extrapolate all candidates to next layer.
  - For each candidate, look for compatible hits.
  - For each candidate, generate a branch with each compatible hit and with the missing hit.
  - Cleanup:
    - Drop candidates with too many missing hits.
    - Drop candidates with bad chi-square.
    - Drop candidates which are subsets of other candidates.
- Final cleanup: select "best" candidate (combination of chi-square and ndf).

Four adaptive filters

- CHA: competition with hard assignment
  - Follows the same basic logic as the CKF.
  - Same cleanup strategy.
  - Modified selection of compatible hits:
    - For each candidate i, compute assignment probabilities of all hits j according to:

$$w_{ij} = \frac{\exp(-\chi^2_{ij}/2)}{\sum_j \exp(-\chi^2_{ij}/2) + \exp(-\chi^2_{\mathrm{cut}}/2)}$$

      where χ^2_{ij} is the compatibility statistics.
    - For the missing hit, we set $\chi^2_{ij} = \chi^2_{\mathrm{cut}}$.
  - Accept only hits with w_{ij} above cut.

Four adaptive filters

- CSA: competition with soft assignment
  - Follows the same basic logic as the CKF.
  - Same cleanup strategy.
  - Modified selection of compatible hits in the same way as before:
    - For each candidate i, compute assignment probabilities of all hits j according to:

$$w_{ij} = \frac{\exp(-\chi^2_{ij}/2)}{\sum_j \exp(-\chi^2_{ij}/2) + \exp(-\chi^2_{\text{cut}}/2)}$$

      where χ^2_{ij} is the compatibility statistics.
    - For the missing hit, we set $\chi^2_{ij} = \chi^2_{\text{cut}}$.
  - Accept only hits with w_{ij} above cut and use w_{ij} as down-weighting factor.

Four adaptive filters

- GSF: Gaussian-sum filter
  - Follows the same basic logic as the CKF.
  - Global competition of all candidates and all hits including the missing hit.
  - Compute new component weights:

$$w_{ij} = \frac{w_i \pi_j \exp(-\chi^2_{ij}/2)}{\sum_{i,j} w_i \pi_j \exp(-\chi^2_{ij}/2)}$$

    where w_i are the old component weights and π_j are the prior probabilities of the hits.
  - Prior probabilites of the hits are uniform, with the exception of the missing hit.
  - Only components with a weight above the cut are accepted.

Four adaptive filters

- DAF: Deterministic annealing filter
  - Iterated Kalman filter with annealing.
  - Initialize all hits with small assignment probabilities.
  - Set starting temperature T.
  - Iterate the following steps:
    - Forward filter, using current assignment probabilities as down-weighting factors.
    - Backward filter, using current assignment probabilities as down-weighting factors.
    - Smoother, i.e. weighted mean of forward and backward filter.

Four adaptive filters

- In each layer k, update of assignment probabilities of all hits j according to:

$$w_j = \frac{\exp(-\chi_j^2/2T)}{\sum_j \exp(-\chi_j^2/2T) + \exp(-\chi_{\mathrm{cut}}^2/2T)}$$

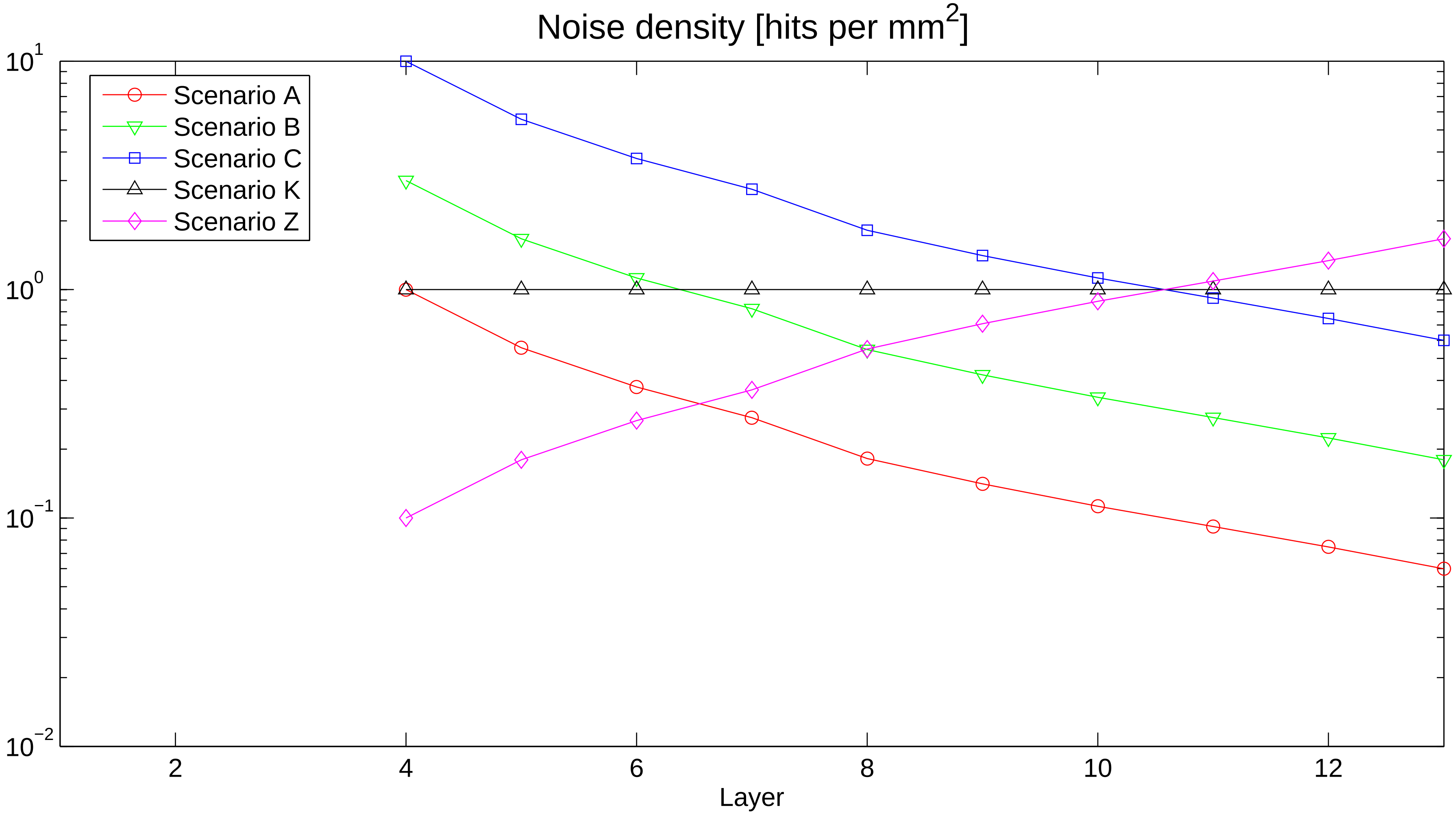
  where χ_j^2 is the compatibility statistics of hit j with respect to the smoothed track.
- Temperature is lowered.

✧ Iteration stops after after a few passes at $T = 1$.

The simulation experiments

- Comparative studies in several simulation experiments
- Experiment 1: "CMS tracker"
  - Three "pixel" layers at $R = 4.3$, 7.2, $11\,\mathrm{cm}$
    $\sigma_{R\Phi} = 0.02\,\mathrm{mm}$, $\sigma_z = 0.02\,\mathrm{mm}$, $X/X_0 = 0.04$
  - Ten "strip" layers at $27 \leq R \leq 110\,\mathrm{cm}$
    $\sigma_{R\Phi} = 0.1\,\mathrm{mm}$, $\sigma_z = 1\,\mathrm{mm}$, $X/X_0 = 0.02$
  - Constant magnetic field, $B_z = 4\,\mathrm{T}$
  - 2500 tracks with $p_T = 10\,\mathrm{GeV}/c$ are simulated.
  - Noise hits are added according to various density assumptions.
  - The true hit is replaced by a noise hit with a probability of $p = 0.1$.

The simulation experiments



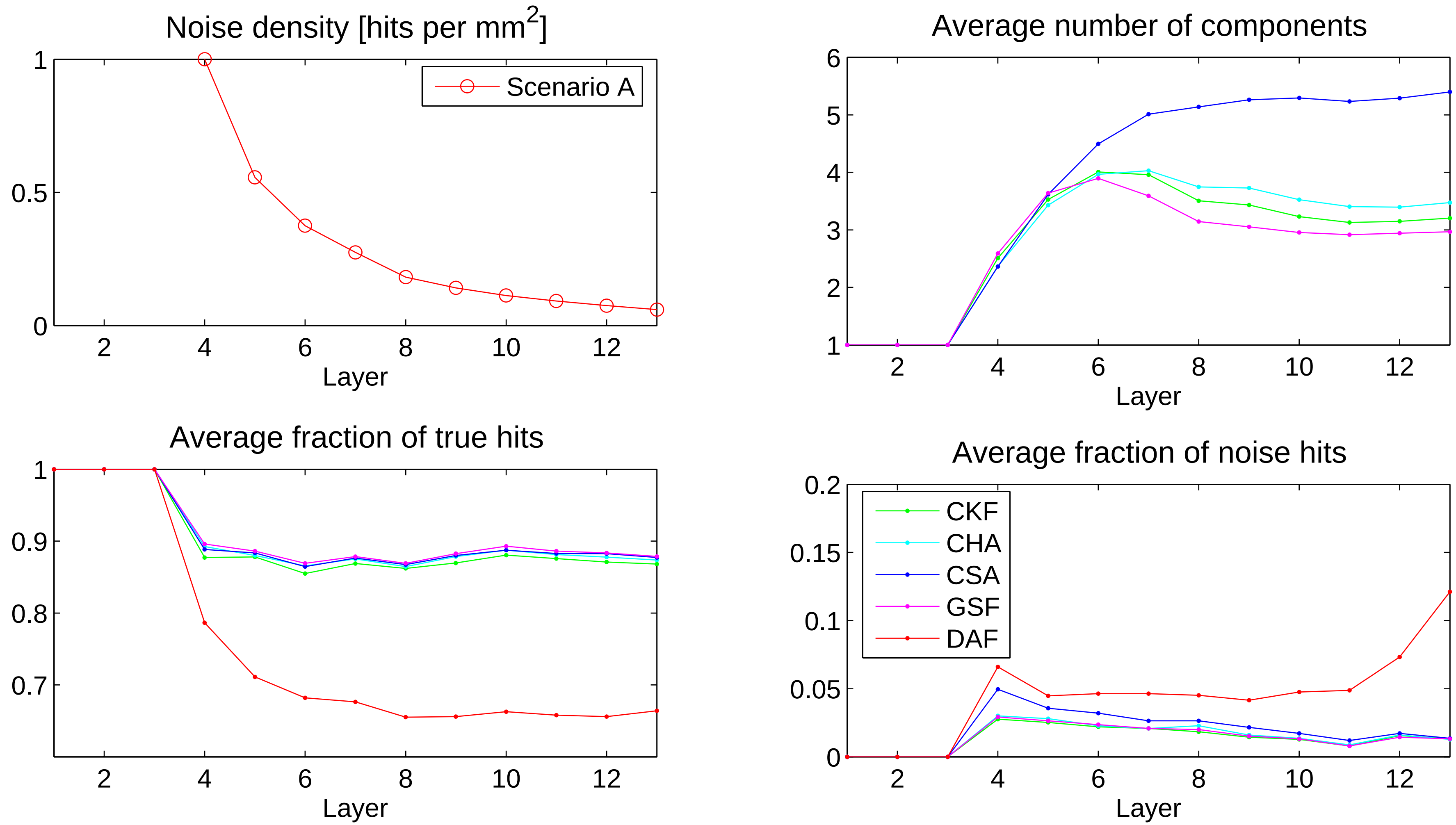
The simulation experiments

- Experiment 2: “ATLAS tracker”
  - Three “pixel” layers at $R = 4.3, 7.2, 11\,\mathrm{cm}$
    $\sigma_{R\Phi} = 0.02\,\mathrm{mm}$, $\sigma_z = 0.02\,\mathrm{mm}$, $X/X_0 = 0.04$
  - Four “strip” layers at $27 \leq R \leq 51.5\,\mathrm{cm}$
    $\sigma_{R\Phi} = 0.1\,\mathrm{mm}$, $\sigma_z = 1\,\mathrm{mm}$, $X/X_0 = 0.02$
  - 36 “drift tube” layers at $55 \leq R \leq 105\,\mathrm{cm}$
    $\sigma_{R\Phi} = 0.2\,\mathrm{mm}$, $\sigma_z = 10\,\mathrm{mm}$, $X/X_0 = 0.005$
  - Constant magnetic field, $B_z = 2\,\mathrm{T}$
  - 2500 tracks with $p_T = 10\,\mathrm{GeV}/c$ are simulated.
  - Noise hits are added with a density of 0.25 per mm^2 in each layer.
  - The true hit is replaced by a noise hit with a probability of $p = 0.2$.

Results

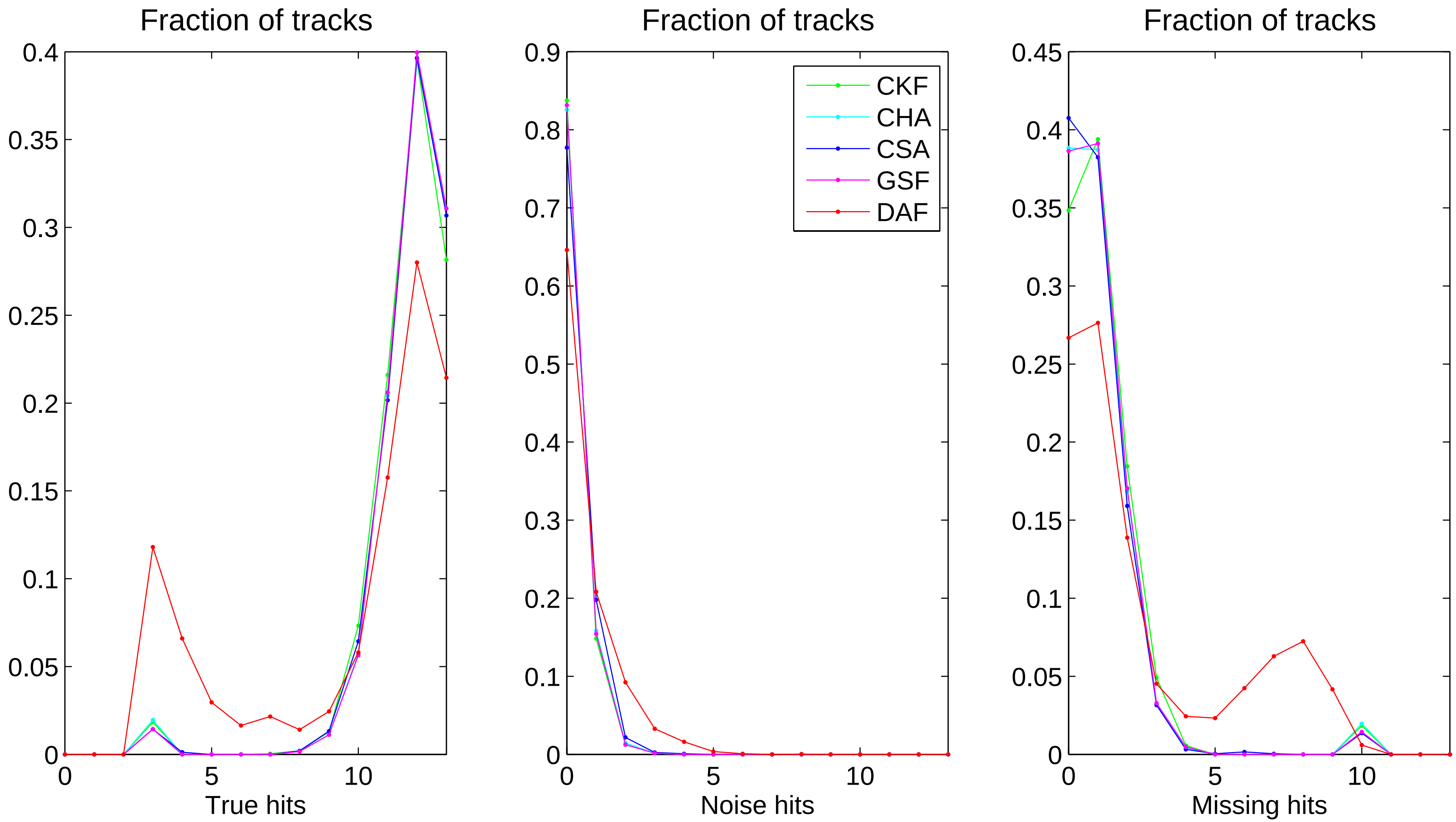
- Experiment 1: “CMS tracker”
  - Perfect seeds from 3 pixel layers
  - Maximum number of candidates: 64
  - Maximum number of missing hits: 4
  - Maximum number of consecutive missing hits: 2
  - Efficiency is fraction of tracks with at least 8 true hits
  - Determination of fake rate:
    - Replace all true hits with noise hits
    - Count number of tracks with more than 8 hits

Results



Noise density [hits per mm^2]
Scenario A
Layer
Average number of components
Layer
Average fraction of true hits
Layer
Average fraction of noise hits
CKF
CHA
CSA
GSF
DAF
Layer

Results



Fraction of tracks
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
True hits
Fraction of tracks
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
Noise hits
CKF
CHA
CSA
GSF
DAF
Fraction of tracks
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
Missing hits

Results

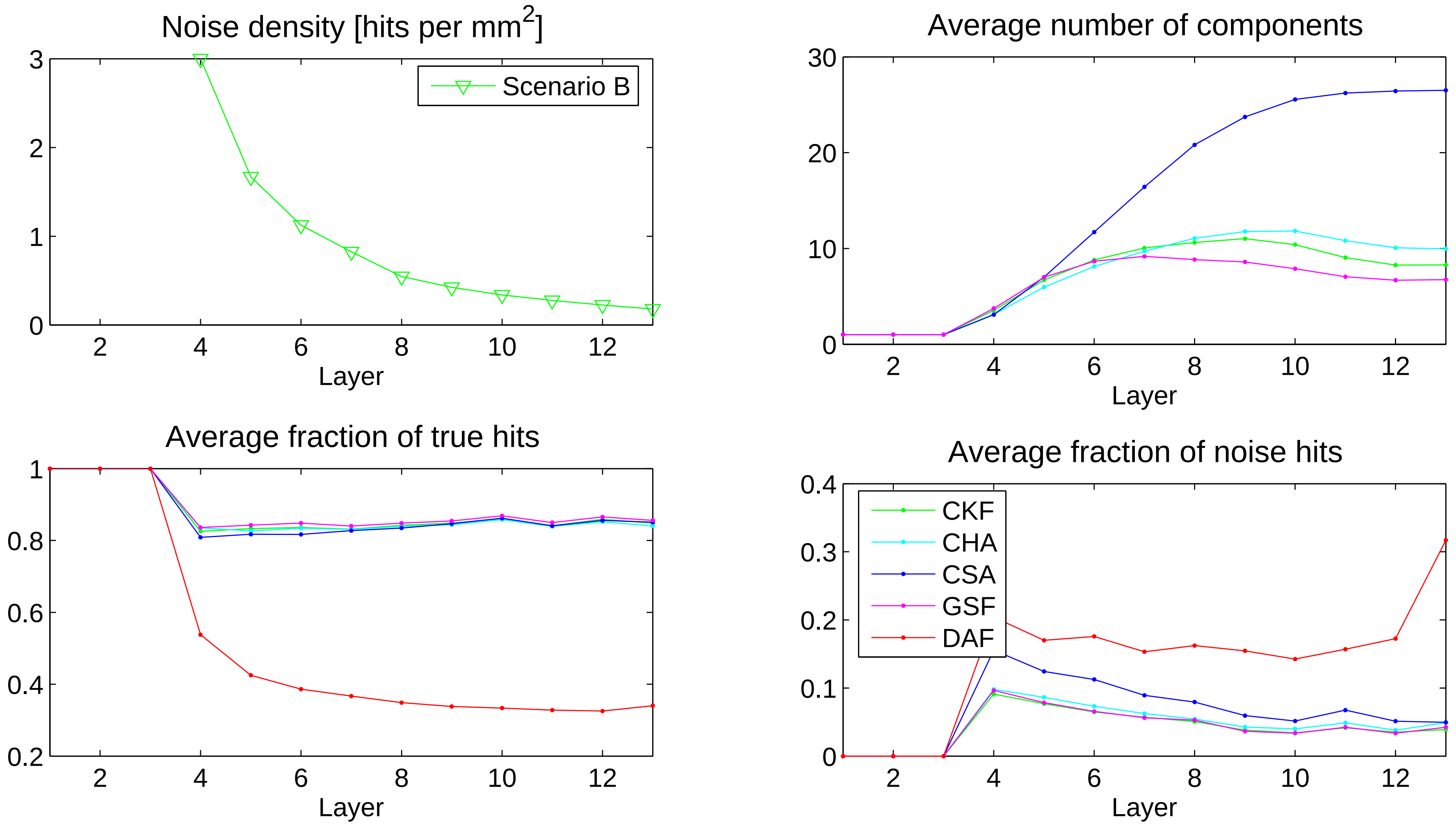
Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	11.7	11.8	11.8	11.8	9.8
Avg. noise hits	0.18	0.19	0.25	0.18	0.58
Avg. missg hits	1.12	1.03	0.96	0.99	2.61
Efficiency	0.98	0.98	0.98	0.99	0.75
Time per seed[1]	1.00	1.04	1.47	0.97	1.07
Fake rate	0.00	0.01	0.02	0.01	0.07
Time per seed[2]	0.49	0.52	0.68	0.55	1.05

Scenario A

[1] True hits are replaced by noise hits with probability $p = 0.1$

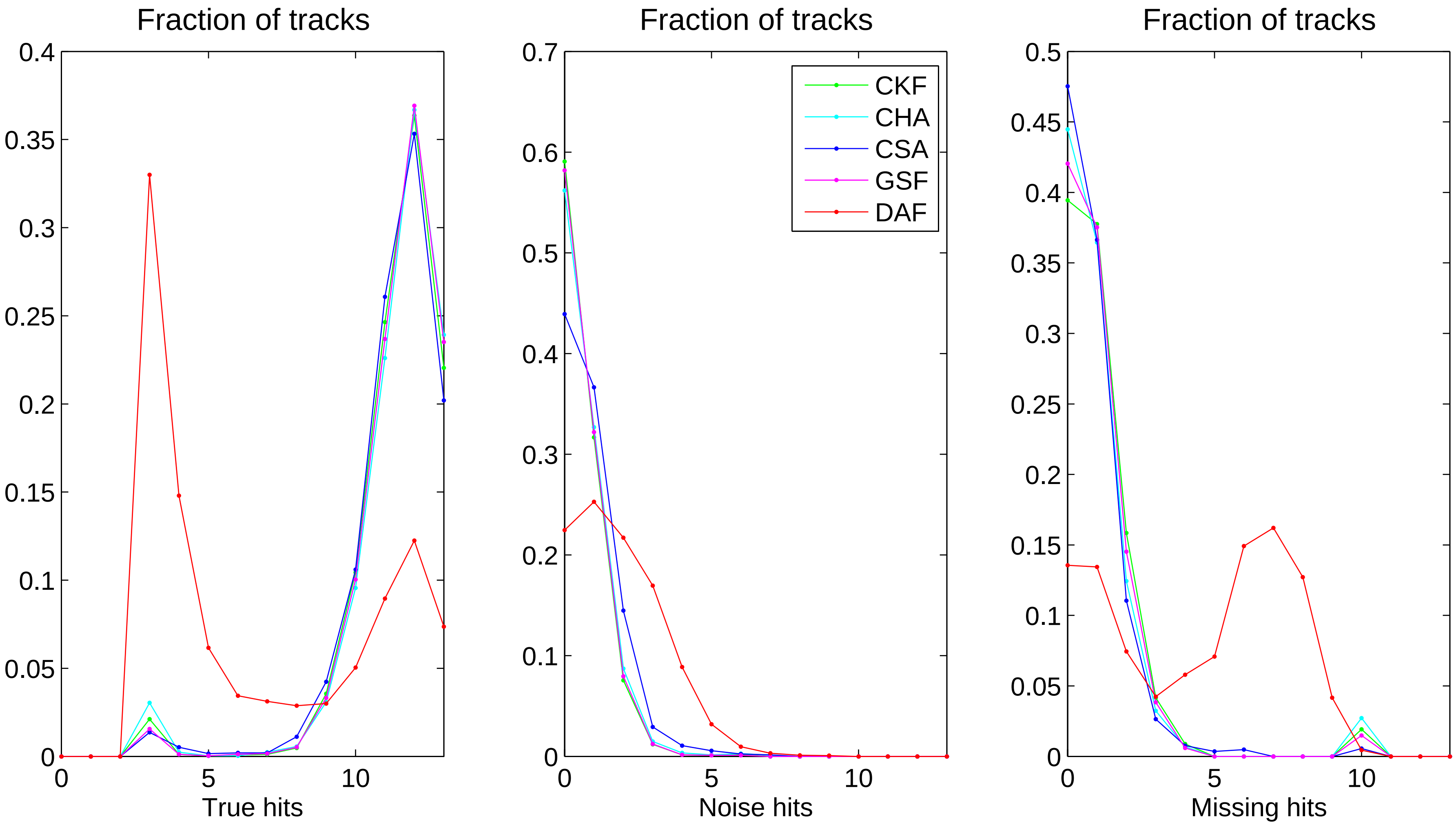
[2] True hits are replaced by noise hits with probability $p = 1$

Results



Noise density [hits per mm^2]
Scenario B
Layer
Average number of components
Layer
Average fraction of true hits
Layer
Average fraction of noise hits
CKF
CHA
CSA
GSF
DAF
Layer

Results



Fraction of tracks
Fraction of tracks
Fraction of tracks
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.5
0.45
0
5
10
True hits
Noise hits
Missing hits
CKF
CHA
CSA
GSF
DAF

Results

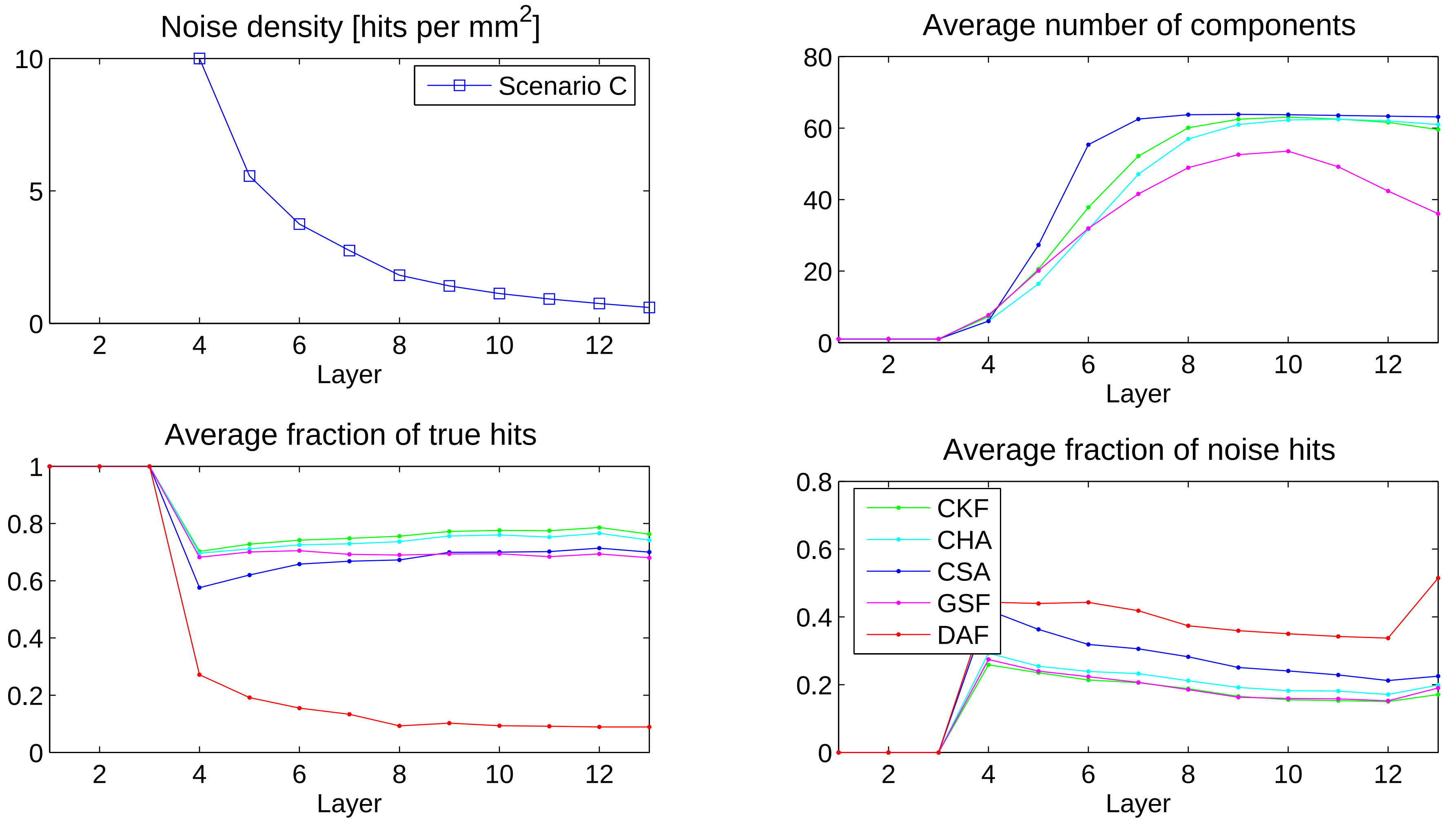
Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	11.4	11.4	11.4	11.5	6.7
Avg. noise hits	0.53	0.59	0.84	0.54	1.81
Avg. missg hits	1.05	1.01	0.80	0.95	4.46
Efficiency	0.98	0.96	0.98	0.98	0.39
Time per seed[1]	1.00	1.11	3.06	0.78	0.39
Fake rate	0.14	0.23	0.46	0.18	0.04
Time per seed[2]	0.53	0.57	1.41	0.54	0.39

Scenario B

[1] True hits are replaced by noise hits with probability $p = 0.1$

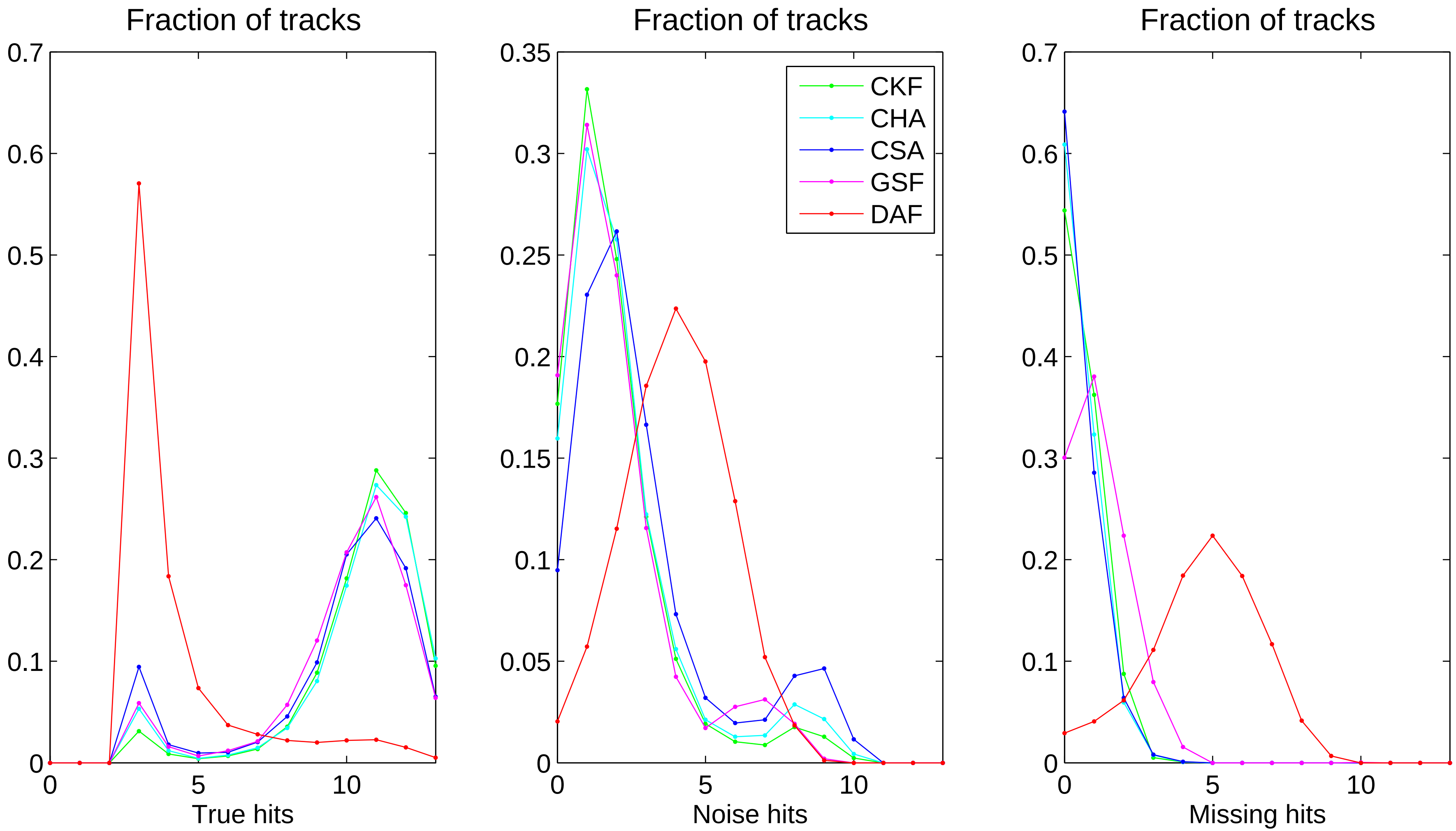
[2] True hits are replaced by noise hits with probability $p = 1$

Results



Noise density [hits per mm^2]
Scenario C
Average number of components
Average fraction of true hits
Average fraction of noise hits
CKF
CHA
CSA
GSF
DAF
Layer

Results


Fraction of tracks
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
True hits
Fraction of tracks
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
Noise hits
CKF
CHA
CSA
GSF
DAF
Fraction of tracks
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
Missing hits

Results

Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	10.5	10.4	9.7	9.9	4.3
Avg. noise hits	1.90	2.16	2.85	1.95	4.02
Avg. missg hits	0.56	0.47	0.44	1.13	4.67
Efficiency	0.94	0.91	0.85	0.89	0.11
Time per seed[1]	1.00	0.96	1.21	0.67	0.04
Fake rate	1.00	1.00	1.00	0.99	0.29
Time per seed[2]	0.86	0.81	1.06	0.56	0.04

Scenario C

[1] True hits are replaced by noise hits with probability $p = 0.1$

[2] True hits are replaced by noise hits with probability $p = 1$

Results

- Improve precision of seed by increasing resolution of the "pixel" hits
- Scenario C1: $\sigma_{R\Phi} = 0.01\,\mathrm{mm}$, $\sigma_z = 0.01\,\mathrm{mm}$
- Scenario C2: $\sigma_{R\Phi} = 0.005\,\mathrm{mm}$, $\sigma_z = 0.005\,\mathrm{mm}$

Results

Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	11.3	11.3	11.1	11.3	5.9
Avg. noise hits	0.94	1.02	1.29	0.94	2.44
Avg. missg hits	0.78	0.68	0.57	0.73	4.70
Efficiency	0.98	0.98	0.98	0.98	0.26
Time per seed[1]	1.00	1.18	3.31	0.60	0.13
Fake rate	0.58	0.72	0.85	0.60	0.13
Time per seed[2]	0.46	0.53	1.83	0.39	0.17

Scenario C1

[1] True hits are replaced by noise hits with probability $p = 0.1$

[2] True hits are replaced by noise hits with probability $p = 1$

Results

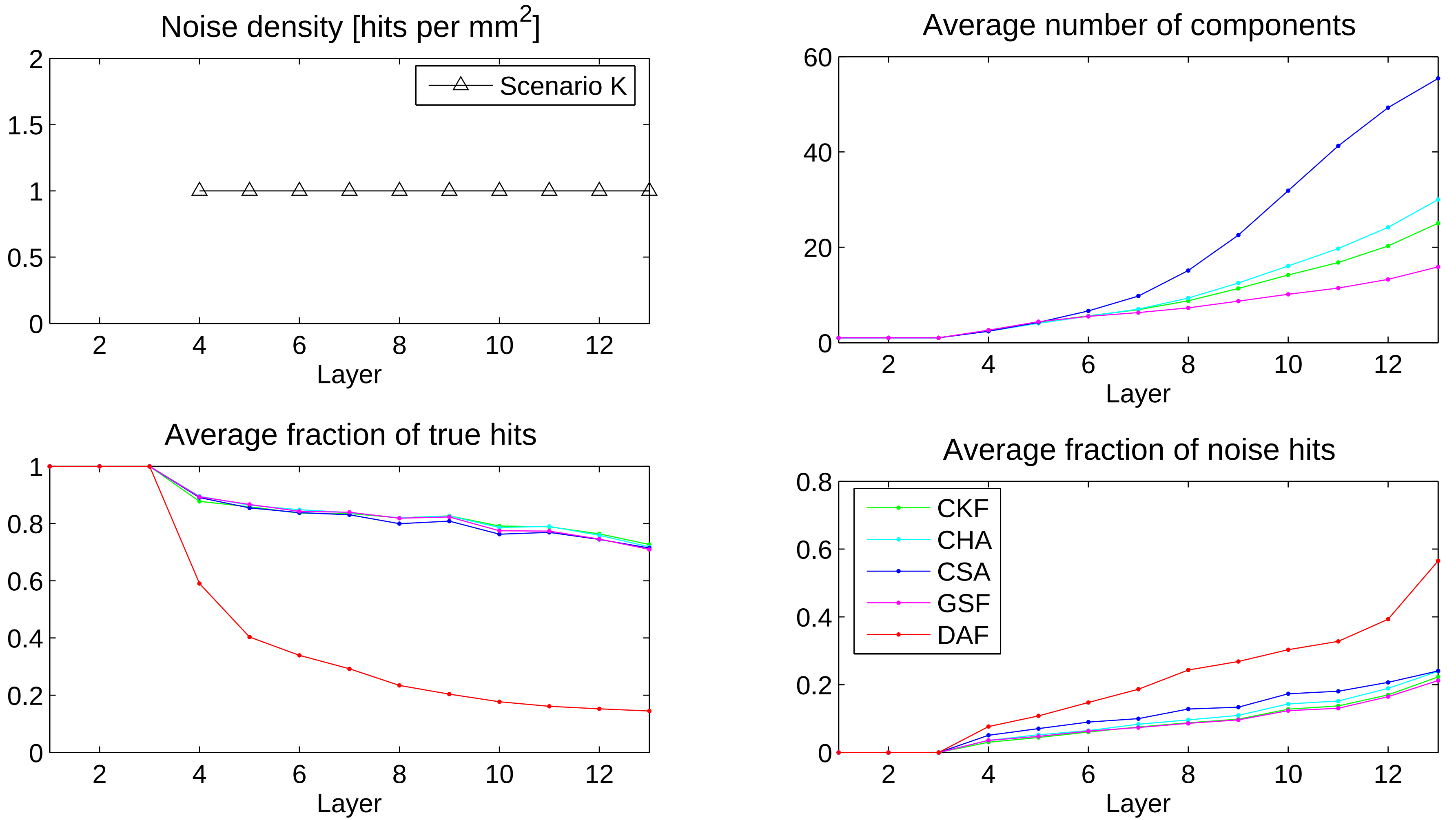
Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	11.5	11.6	11.6	11.6	8.6
Avg. noise hits	0.52	0.58	0.73	0.55	1.23
Avg. missg hits	0.99	0.86	0.71	0.84	3.20
Efficiency	0.98	0.98	0.99	0.98	0.61
Time per seed[1]	1.00	1.17	2.78	0.77	0.55
Fake rate	0.14	0.25	0.47	0.22	0.13
Time per seed[2]	0.34	0.39	0.82	0.36	0.40

Scenario C2

[1] True hits are replaced by noise hits with probability $p = 0.1$

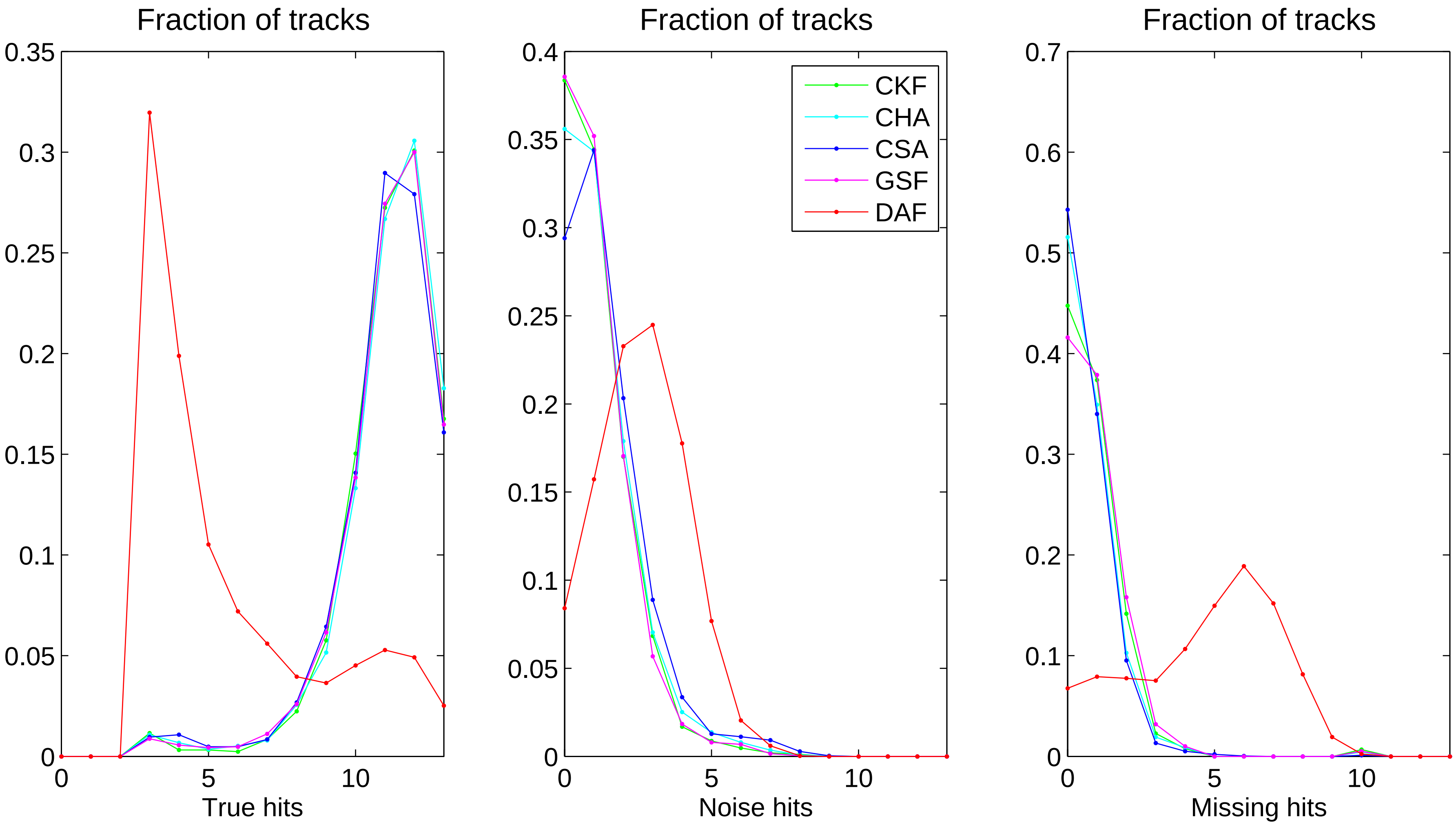
[2] True hits are replaced by noise hits with probability $p = 1$

Results



Noise density [hits per mm^2]
Scenario K
Layer
Average number of components
Layer
Average fraction of true hits
Layer
Average fraction of noise hits
CKF
CHA
CSA
GSF
DAF
Layer

Results



Fraction of tracks
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
True hits
Fraction of tracks
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
Noise hits
CKF
CHA
CSA
GSF
DAF
Fraction of tracks
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
Missing hits

Results

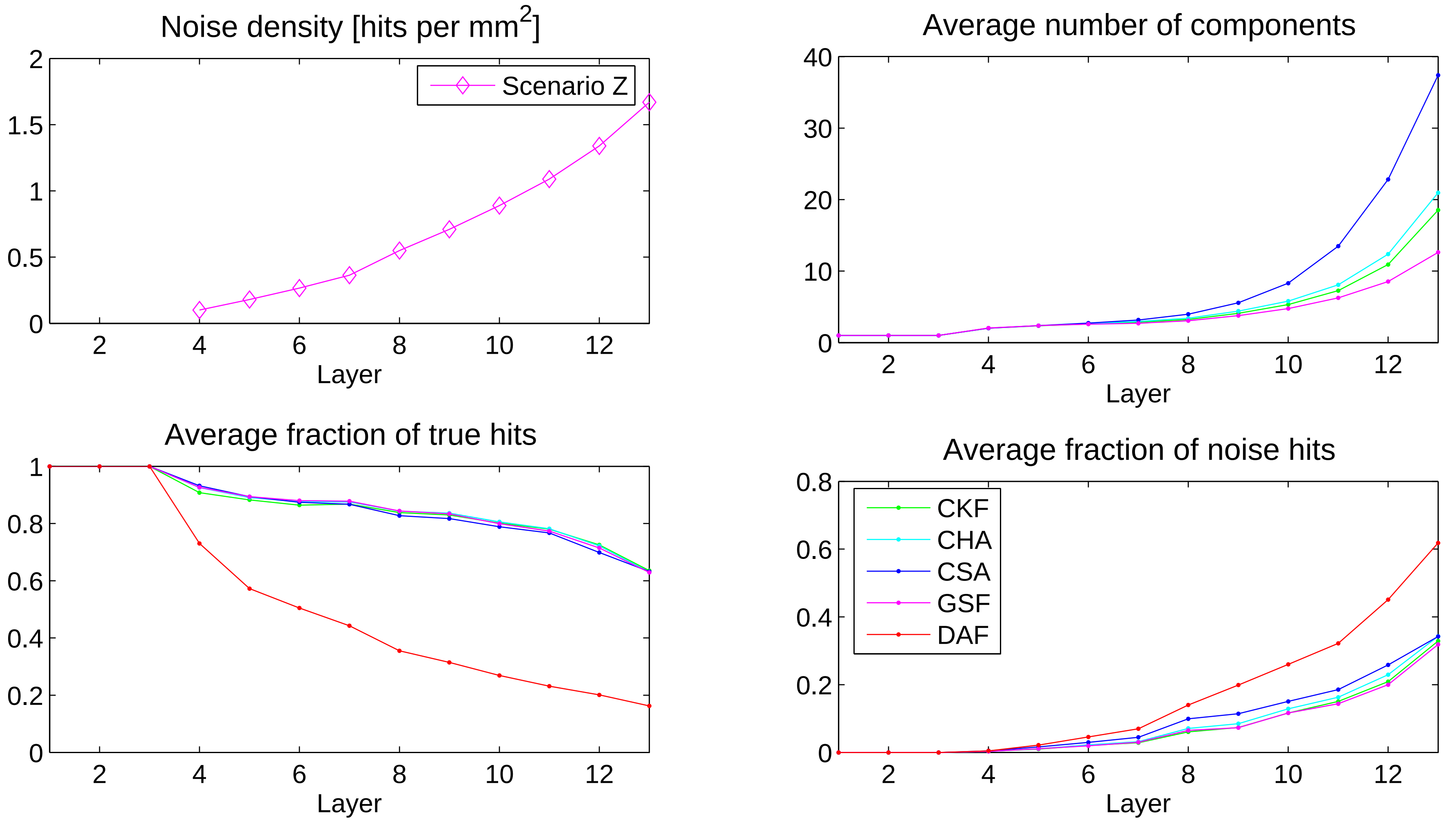
Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	11.1	11.1	11.0	11.1	5.7
Avg. noise hits	1.05	1.16	1.37	1.03	2.62
Avg. missg hits	0.82	0.69	0.61	0.88	4.68
Efficiency	0.97	0.97	0.96	0.97	0.25
Time per seed[1]	1.00	1.17	2.69	0.66	0.24
Fake rate	0.77	0.88	0.94	0.77	0.08
Time per seed[2]	0.51	0.64	1.91	0.43	0.24

Scenario K

[1] True hits are replaced by noise hits with probability $p = 0.1$

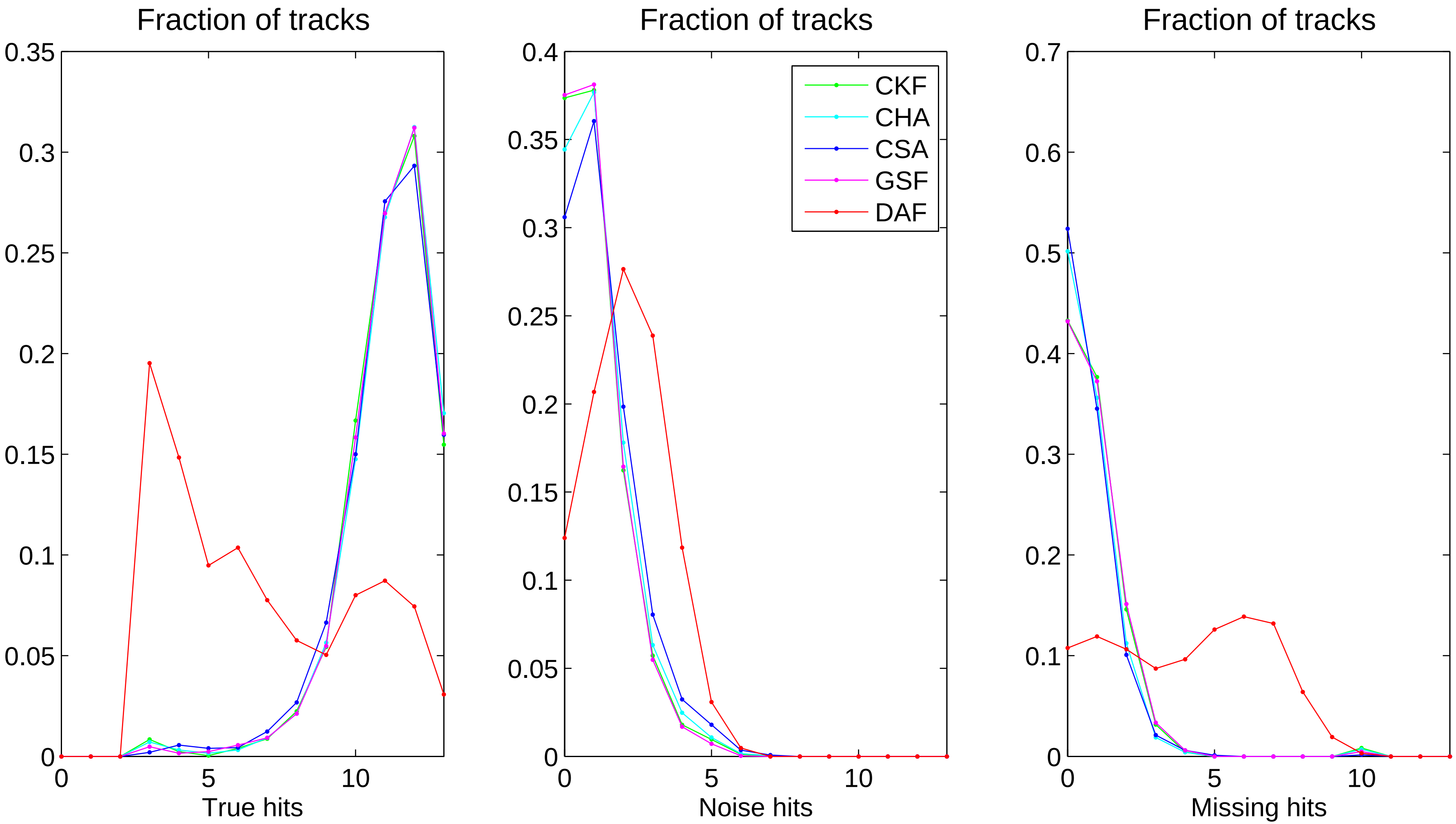
[2] True hits are replaced by noise hits with probability $p = 1$

Results



Noise density [hits per mm^2]
Scenario Z
2
1.5
1
0.5
0
2
4
6
8
10
12
Layer
Average number of components
40
30
20
10
0
2
4
6
8
10
12
Layer
Average fraction of true hits
1
0.8
0.6
0.4
0.2
0
2
4
6
8
10
12
Layer
Average fraction of noise hits
0.8
0.6
0.4
0.2
0
CKF
CHA
CSA
GSF
DAF
2
4
6
8
10
12
Layer

Results



Fraction of tracks
True hits
Fraction of tracks
Noise hits
CKF
CHA
CSA
GSF
DAF
Fraction of tracks
Missing hits

Results

Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	11.1	11.2	11.1	11.2	6.8
Avg. noise hits	1.00	1.09	1.25	0.98	2.13
Avg. missg hits	0.87	0.73	0.66	0.85	4.08
Efficiency	0.98	0.98	0.97	0.98	0.39
Time per seed[1]	1.00	1.11	2.03	0.80	0.47
Fake rate	0.35	0.52	0.67	0.45	0.05
Time per seed[2]	0.26	0.36	1.13	0.31	0.47

Scenario Z

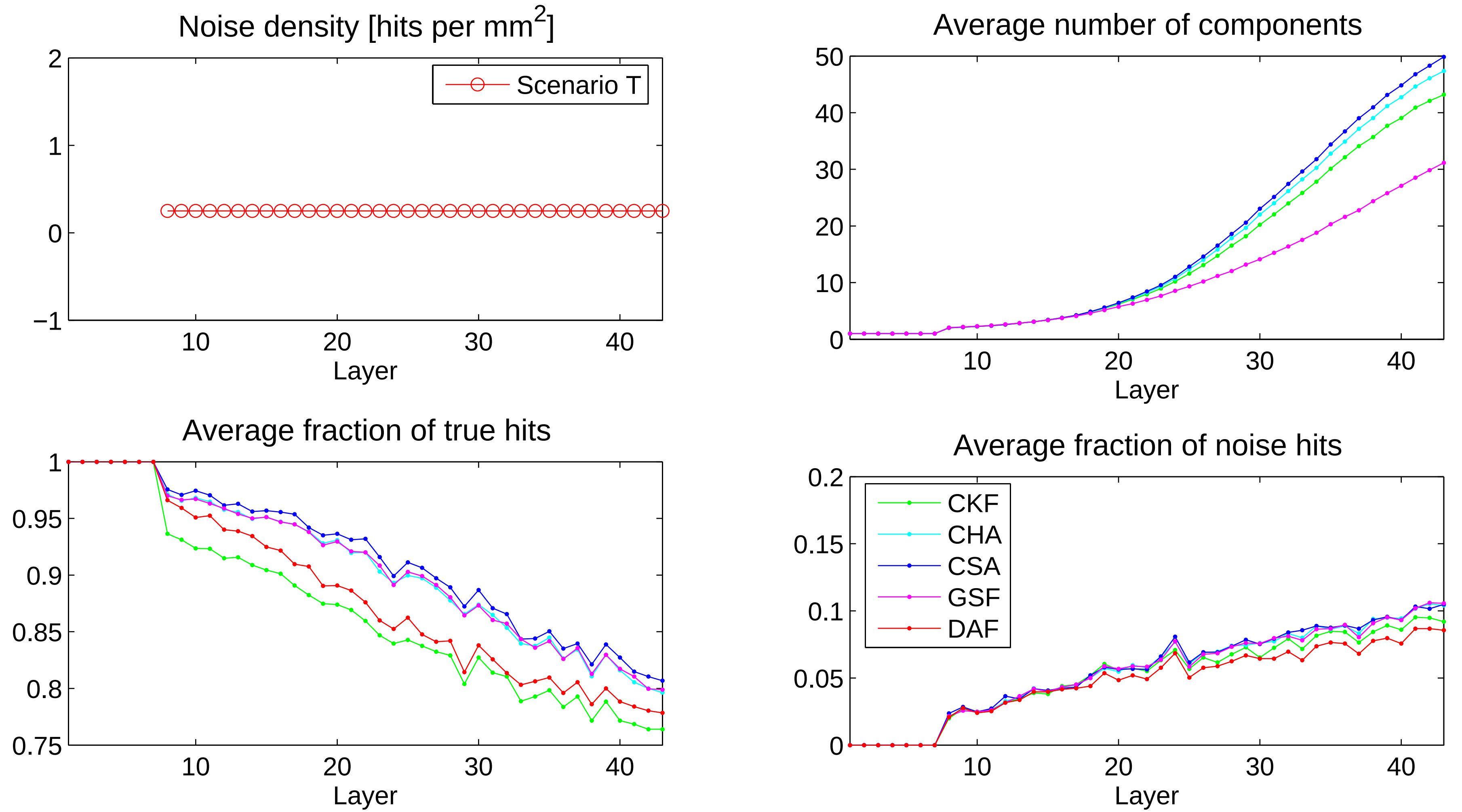
[1] True hits are replaced by noise hits with probability $p = 0.1$

[2] True hits are replaced by noise hits with probability $p = 1$

Results

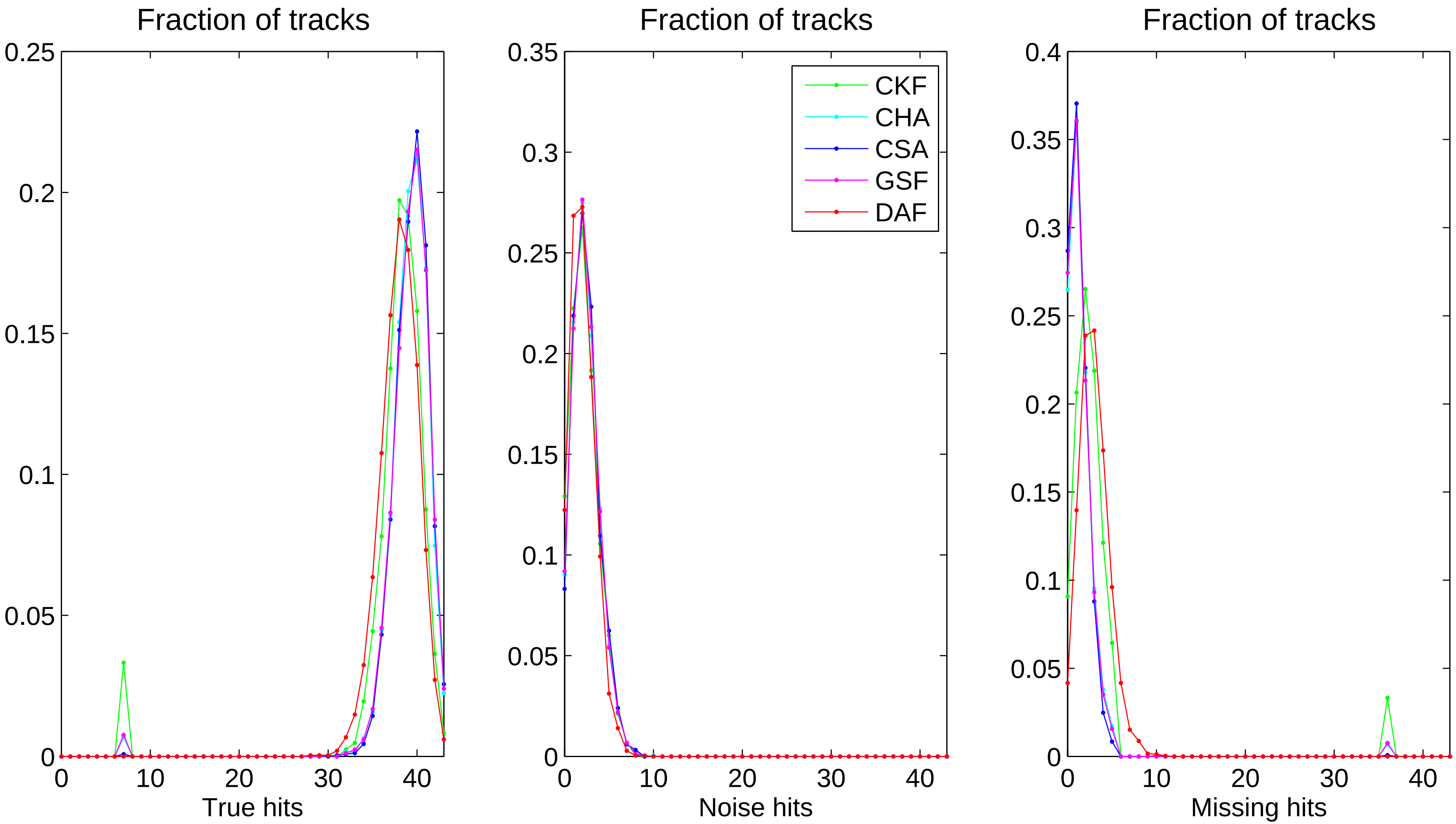
- Experiment 2: "ATLAS tracker"
  - Perfect seeds from 7 silicon (pixel+strip) layers
  - Maximum number of candidates: 64
  - Maximum number of missing hits: 5
  - Maximum number of consecutive missing hits: 3
  - Efficiency is fraction of tracks with at least 25 true hits
  - Determination of fake rate:
    - Replace all true hits with noise hits
    - Count number of tracks with more than 36 hits

Results



Noise density [hits per mm^2]
Scenario T
2
1
0
−1
10
20
30
40
Layer
Average number of components
50
40
30
20
10
0
10
20
30
40
Layer
Average fraction of true hits
1
0.95
0.9
0.85
0.8
0.75
10
20
30
40
Layer
Average fraction of noise hits
0.2
0.15
0.1
0.05
0
CKF
CHA
CSA
GSF
DAF
10
20
30
40
Layer

Results



Fraction of tracks
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
True hits
Fraction of tracks
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
Noise hits
CKF
CHA
CSA
GSF
DAF
Fraction of tracks
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
Missing hits

Results

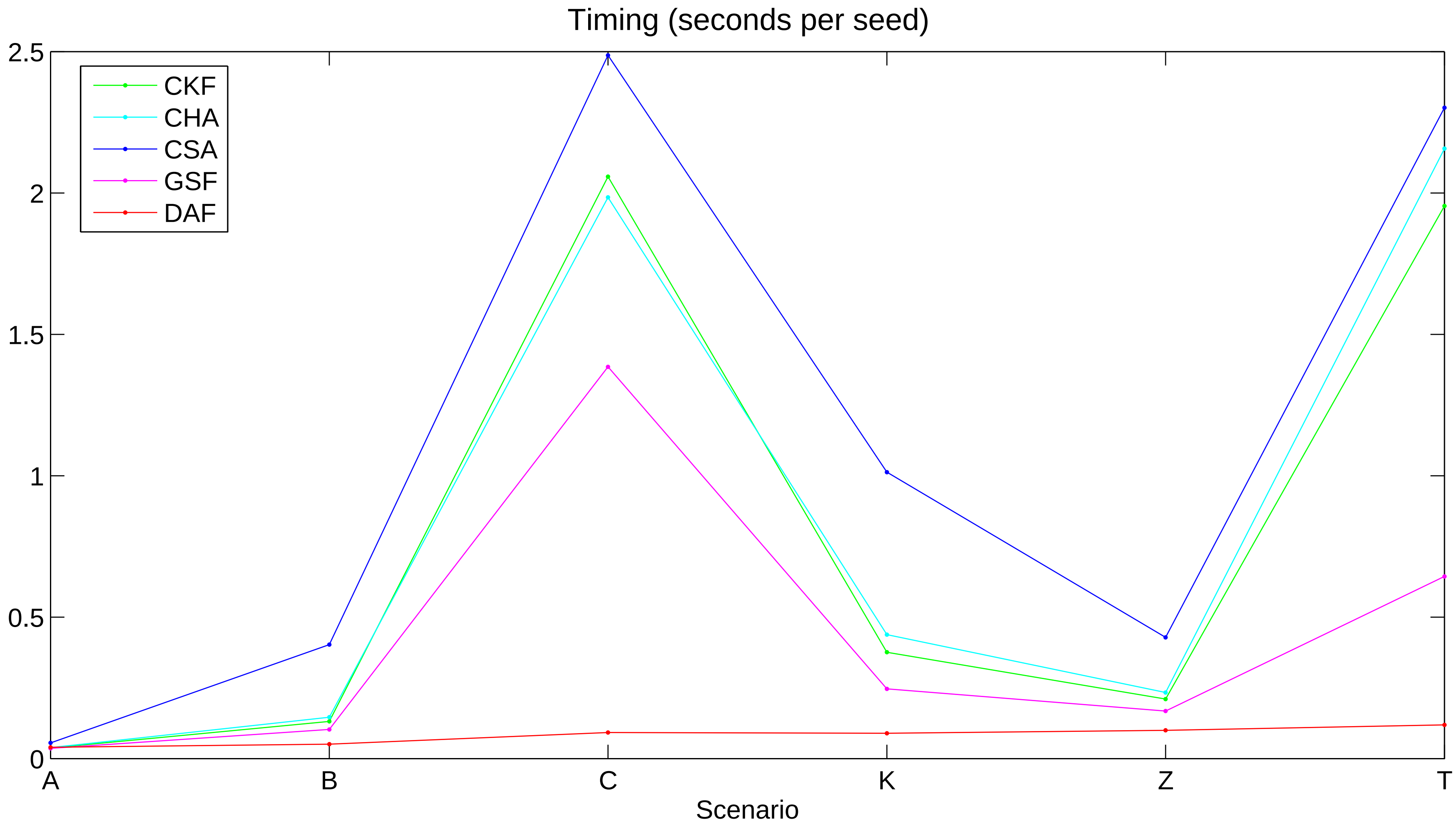
Method	CKF	CHA	CSA	GSF	DAF
Avg. true hits	37.4	39.1	39.4	39.1	38.0
Avg. noise hits	2.23	2.36	2.39	2.35	2.04
Avg. missg hits	3.40	1.56	1.25	1.56	2.97
Efficiency	0.97	0.99	1.00	0.99	1.00
Time per seed[1]	1.00	1.10	1.18	0.33	0.06
Fake rate	0.01	0.03	0.54	0.03	0.02
Time per seed[2]	0.14	0.16	0.64	0.08	0.06

Scenario T

[1] True hits are replaced by noise hits with probability $p = 0.2$

[2] True hits are replaced by noise hits with probability $p = 1$

Results



Timing (seconds per seed)
CKF
CHA
CSA
GSF
DAF
2.5
2
1.5
1
0.5
0
A
B
C
K
Z
T
Scenario

Discussion and outlook

- We have investigated four adaptive filters and compared them to the combinatorial Kalman filter.
- In the "CMS tracker" setup, the DAF fails because of insufficient quality of the seeds.
- The CKF and the GSF have about the same performance, but the GSF is somewhat faster.
- Prior knowledge about the noise distribution can be incorporated into the GSF.
- In noise scenario C, all methods break down, as it is no longer possible to distinguish correct from wrong seeds. This can be cured by increasing the precision of the seeds.

Discussion and outlook

- In the "ATLAS tracker" setup, all but one filters achieve full efficiency at low fake rate. The DAF is by far the fastest, more than 15 times faster than the CKF.
- Future work:
  - Introduce noise also into seeds
  - More realistic noise generation, including hit merging
  - Investigate sensitivity of GSF on prior assumptions
  - Compare with other concepts, e.g. Kisel's cellular automaton