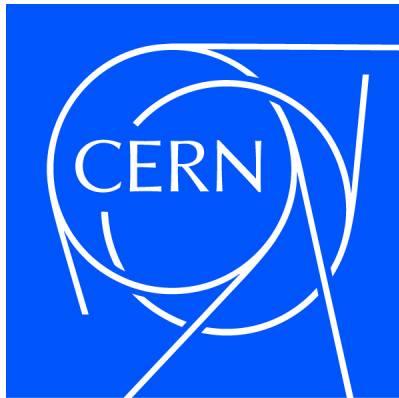


Track finding in HEP

Christoph Smaczny

06.11.2015



NETZWERK
TEILCHENWELT

Track finding in HEP

- Eingrenzung der Aufgabe
- Generieren von Testdaten
- Track finding: Ein erster Ansatz
- Auswertung / Probleme
- Ausblick: Weiterführende Ansätze

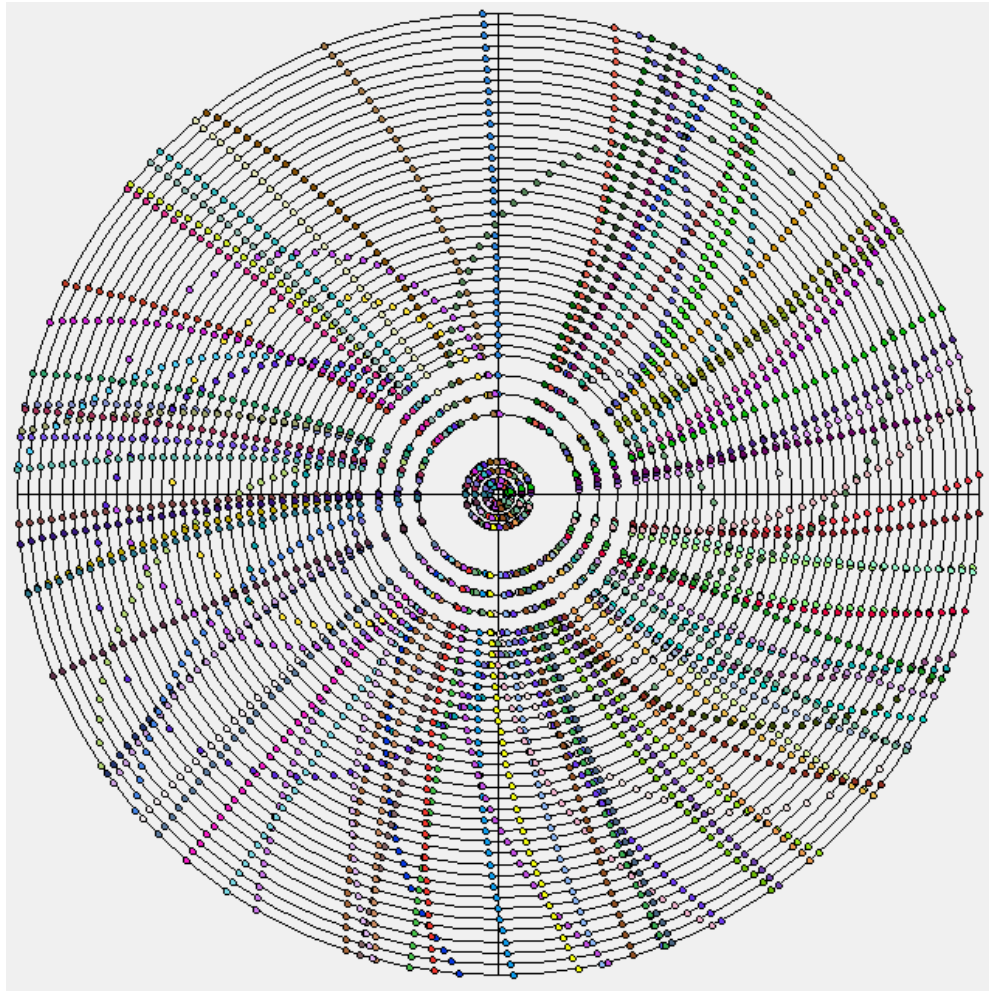
Eingrenzung der Aufgabe

- Um Events weiterverarbeiten zu können, müssen Hits in Impuls- und Ortswerte umgewandelt werden
- 2 Schritte:
 - Track reconstruction
 - Track finding
 - Track fitting
 - Vertex reconstruction
- In diesem Projekt nur 1. Teil des 1. Schritts: Welche Hits könnten Track bilden?

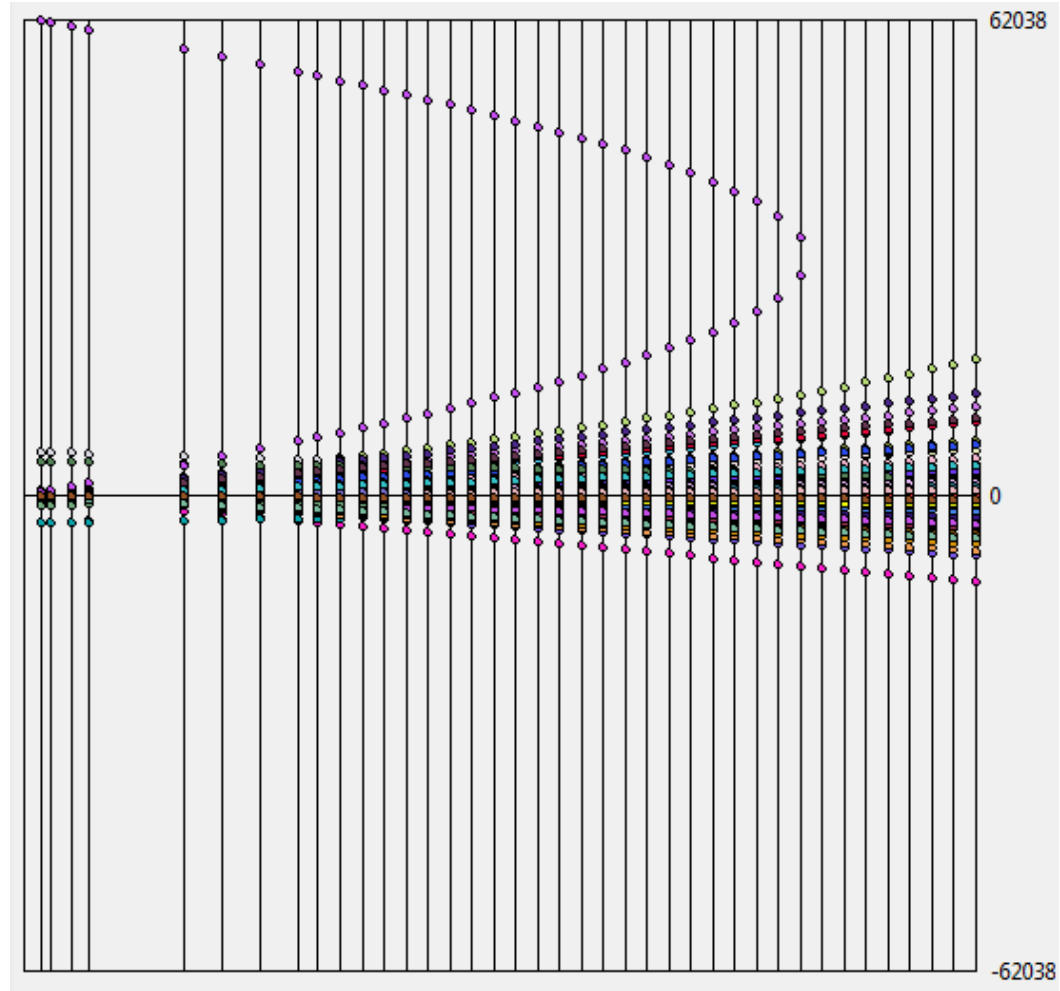
Generierung von Testdaten

- Koordinatensystem: z-Achse entlang Strahlachse, xy-Ebene senkrecht dazu
- Für jedes Teilchen:
 - Beginne im Zentrum des Detektors
 - Solange nicht auf äußerstem Detektorzylinder:
 - Berechne in xy-Ebene Schnittpunkte der Flugbahn mit nächstem Detektorzylinder
 - Falls Schnittpunkt(e) existiert:
 - wähle richtigen Schnittpunkt
 - Berechne z-Koordinate des Schnittpunkts
 - Füge Hit aktuellem Track hinzu (verändere Hitposition auf Detektor zuvor normalverteilt zufällig leicht)
 - Der nächste Zylinder wird zum aktuellen
 - Sonst: gehe von diesem Zylinder aus wieder Zylinderweise nach innen und füge entsprechende Hits hinzu
 - Füge Track der Trackliste hinzu

Generierung von Testdaten



100 Tracks



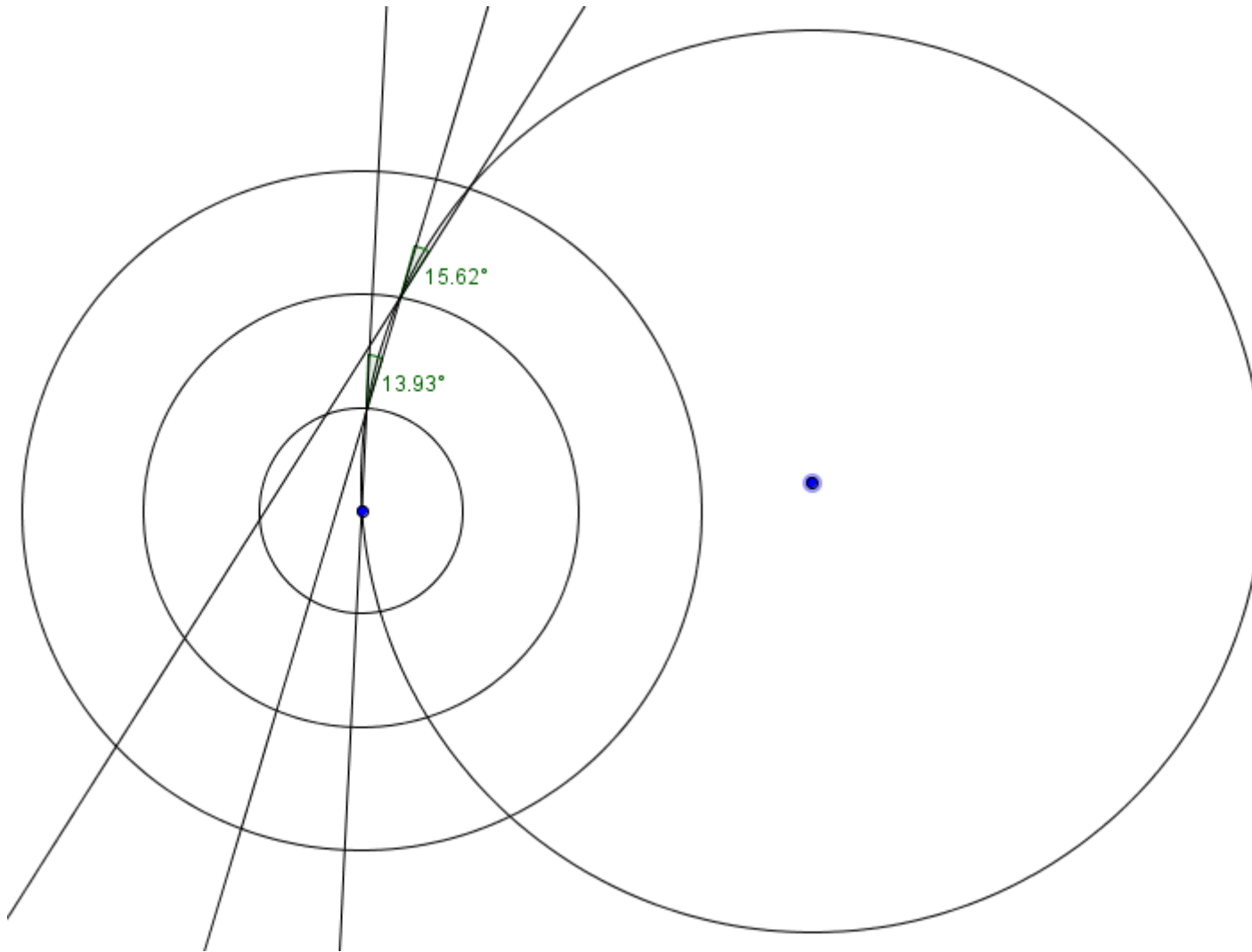
Track finding: Ein erster Ansatz

- Eingabe: Liste von Hits (Position, Detektorzylinder)
- Erzeuge Array, das für jeden Detektorzylinder Liste seiner Hits enthält
- Beginne im Zentrum des Detektors
- TrackCandidates := []
- Track := []
- A: Für jeden Hit auf aktuellem Zylinder: //Tiefentraversierung
 - Falls Winkelkriterium und Steigungskriterium erfüllt:
 - Wenn aktueller Zylinder äußerster Zylinder: Füge Track zu TrackCandidates hinzu
 - Sonst:
 - Für Track = Track+Hit und nächsten Zylinder zu A
 - Track hat danach wieder alten Wert (ohne aktuellen Hit)

Track finding: Ein erster Ansatz

- Winkelkriterium:
 - Gegeben 3 Punkte P1, P2, P3 mit $P1.layer + 2 = P2.layer + 1 = P3.layer$
 - $\gamma :=$ (kleinerer) Winkel zwischen den Geraden P1P2 und P2P3
 - $\langle \gamma \rangle :=$ Mittelwert der bisherigen γ
 - $|\gamma - \langle \gamma \rangle| < \alpha/2$
- Steigungskriterium:
 - Gegeben 2 Punkte P1, P2 mit $P1.layer + 1 = P2.layer$
 - Differenz der Steigungen in z-Richtung zwischen Gerade P1P2 und mittlerer Steigung kleiner als ein Schwellwert
- Sind noch zu wenige Punkte zur Anwendung der Kriterien da, werde sie immer aufgenommen

Track finding: Ein erster Ansatz

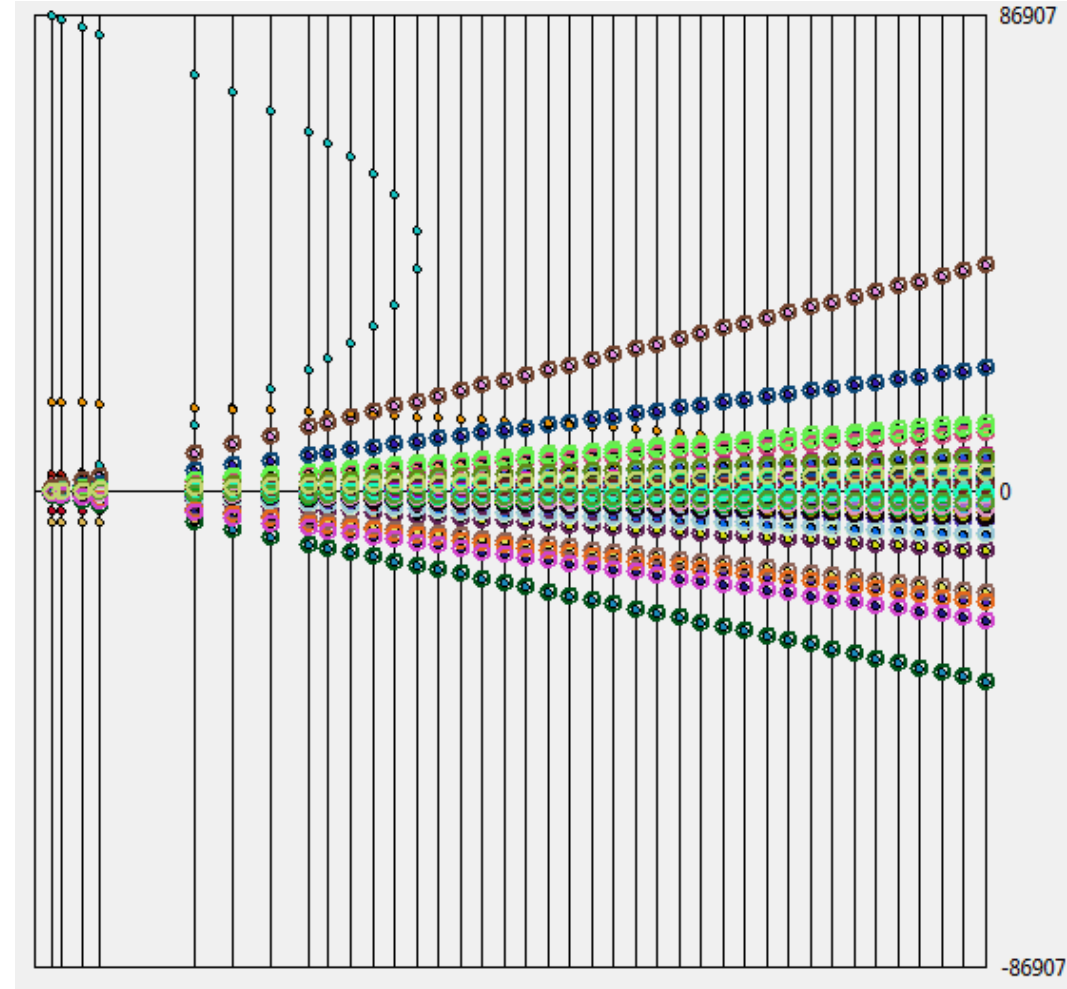
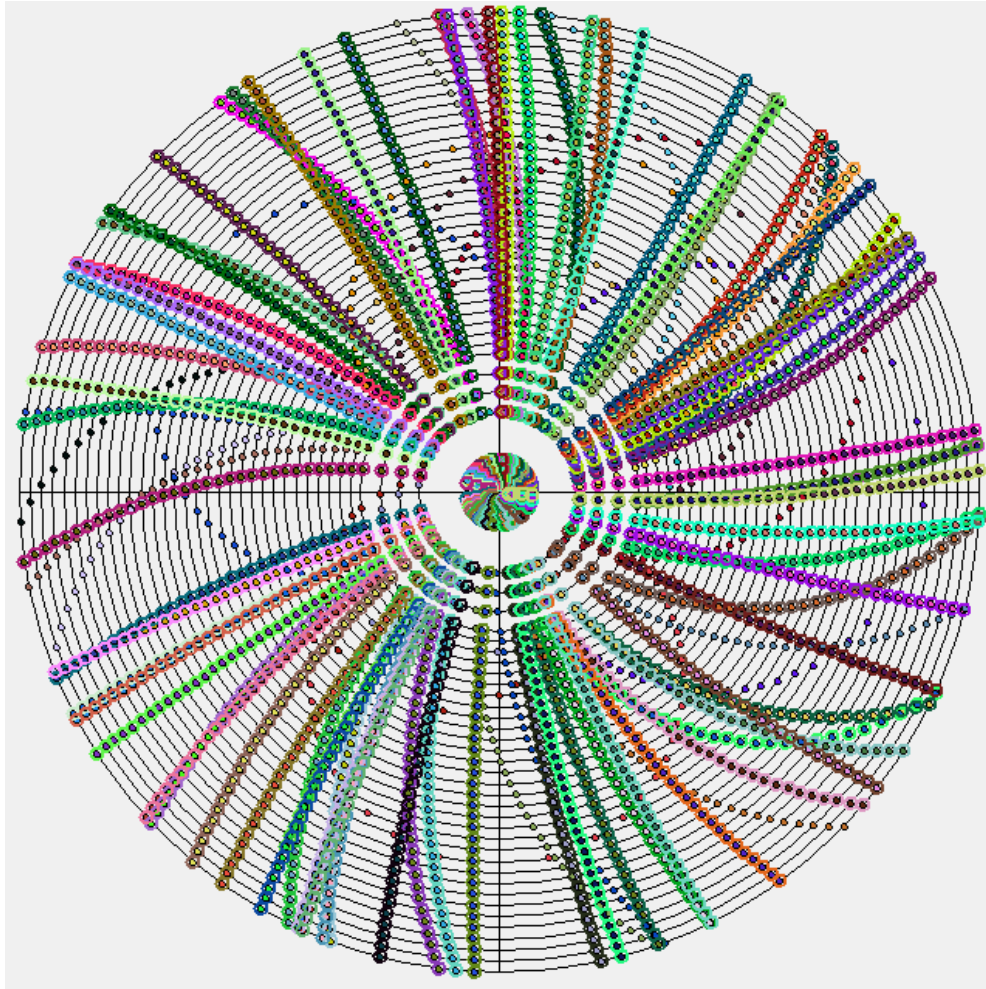


Winkel nur für
große
Trackradien
ungefähr gleich
→
unproblematisch,
da meist Teilchen
mit großen
Impulsen von
Interesse sind

Gütekriterien für Track finding

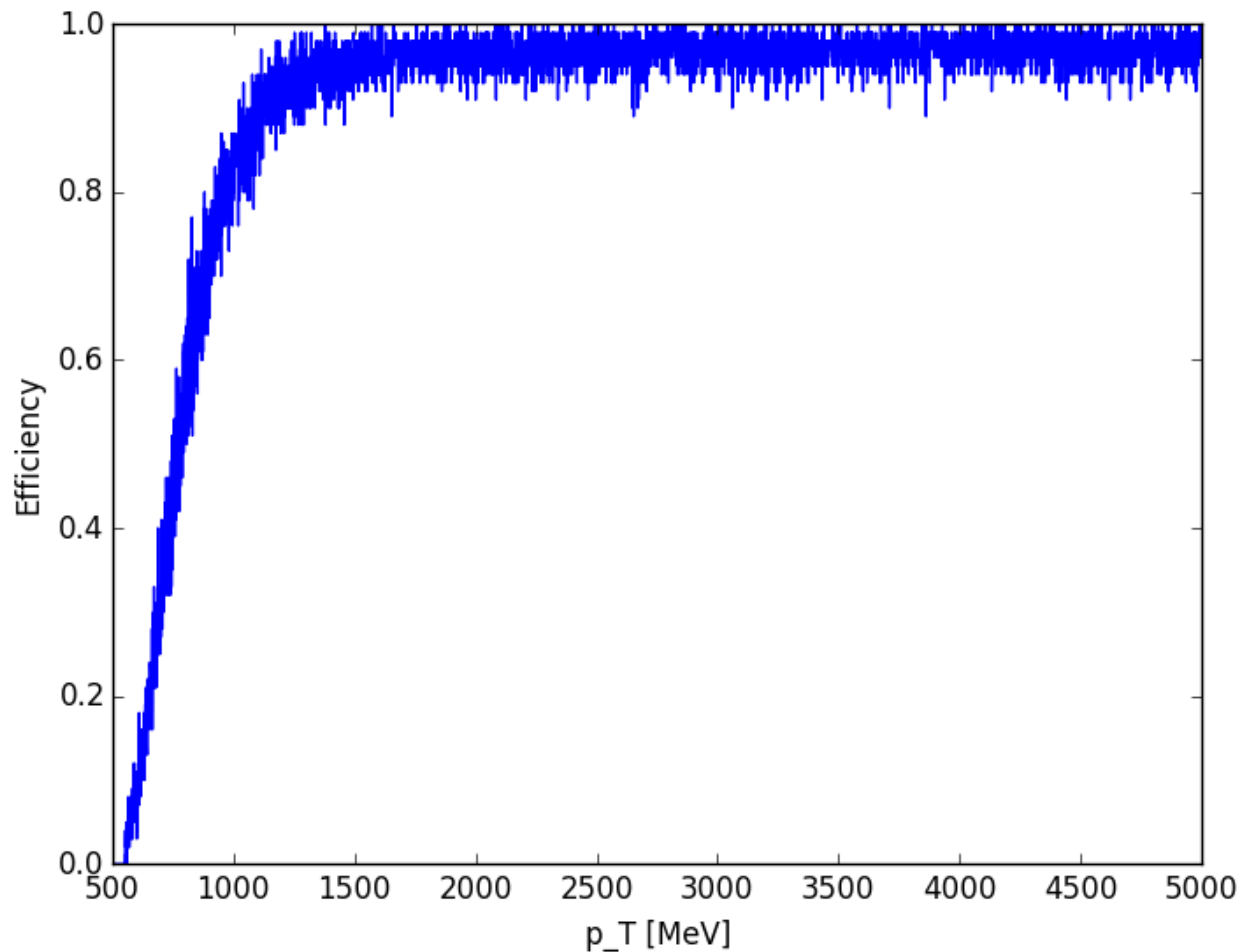
- Efficiency: Anteil der vollständig erkannte Tracks an den generierte Tracks
- Fakerate: Anteil falscher Tracks an allen gefundenen Tracks

Beispielausgabe



Efficiency: 86 %, Fakerate: 19 %

Transversalimpuls \rightarrow Efficiency

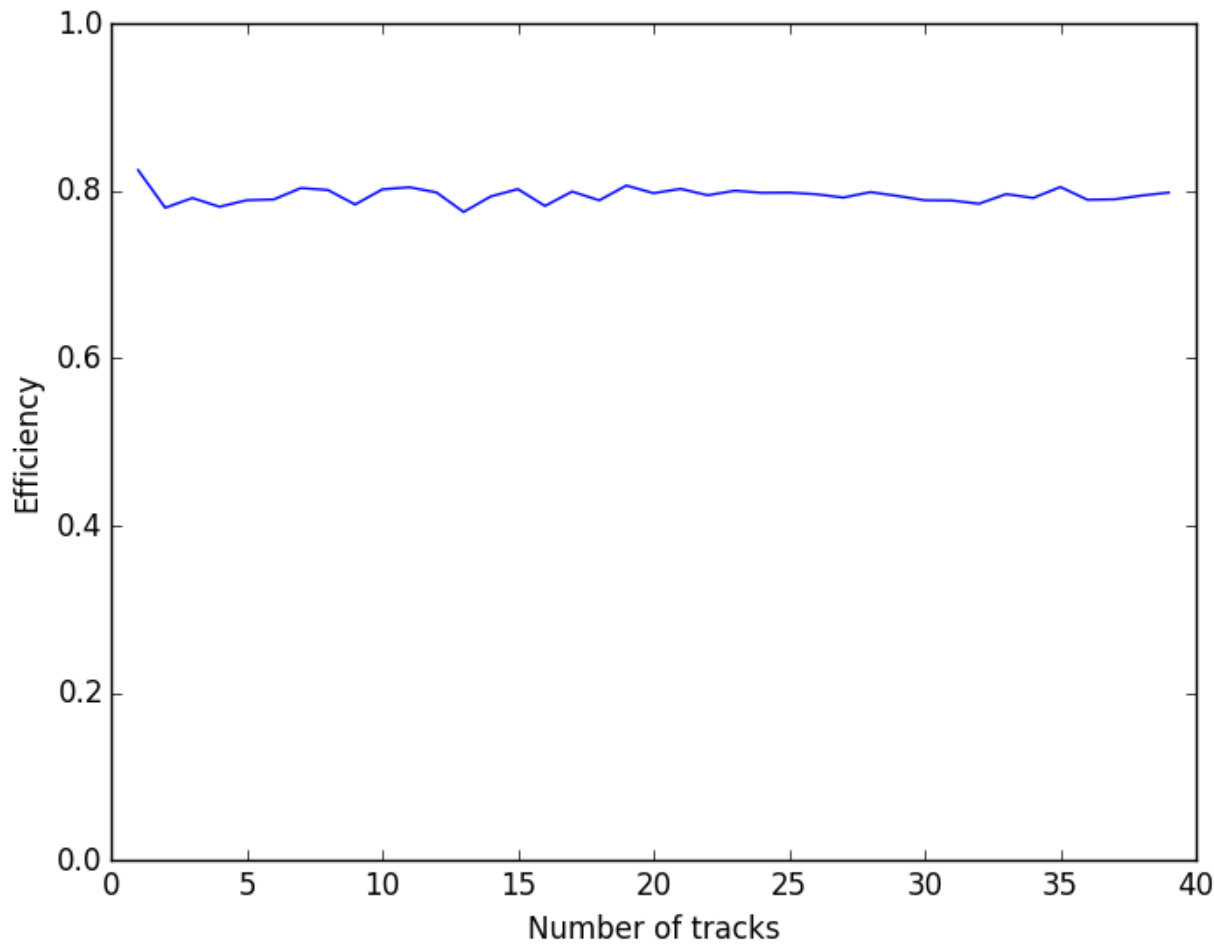


Zufällige
Verschiebung für
xy und z:
Normalverteilt mit
 $\sigma = 1$
(Längen in mm)

Gemittelt über 100
Durchläufe

$$\alpha = 2\pi/40 = 0.157$$

Anzahl Tracks → Efficiency

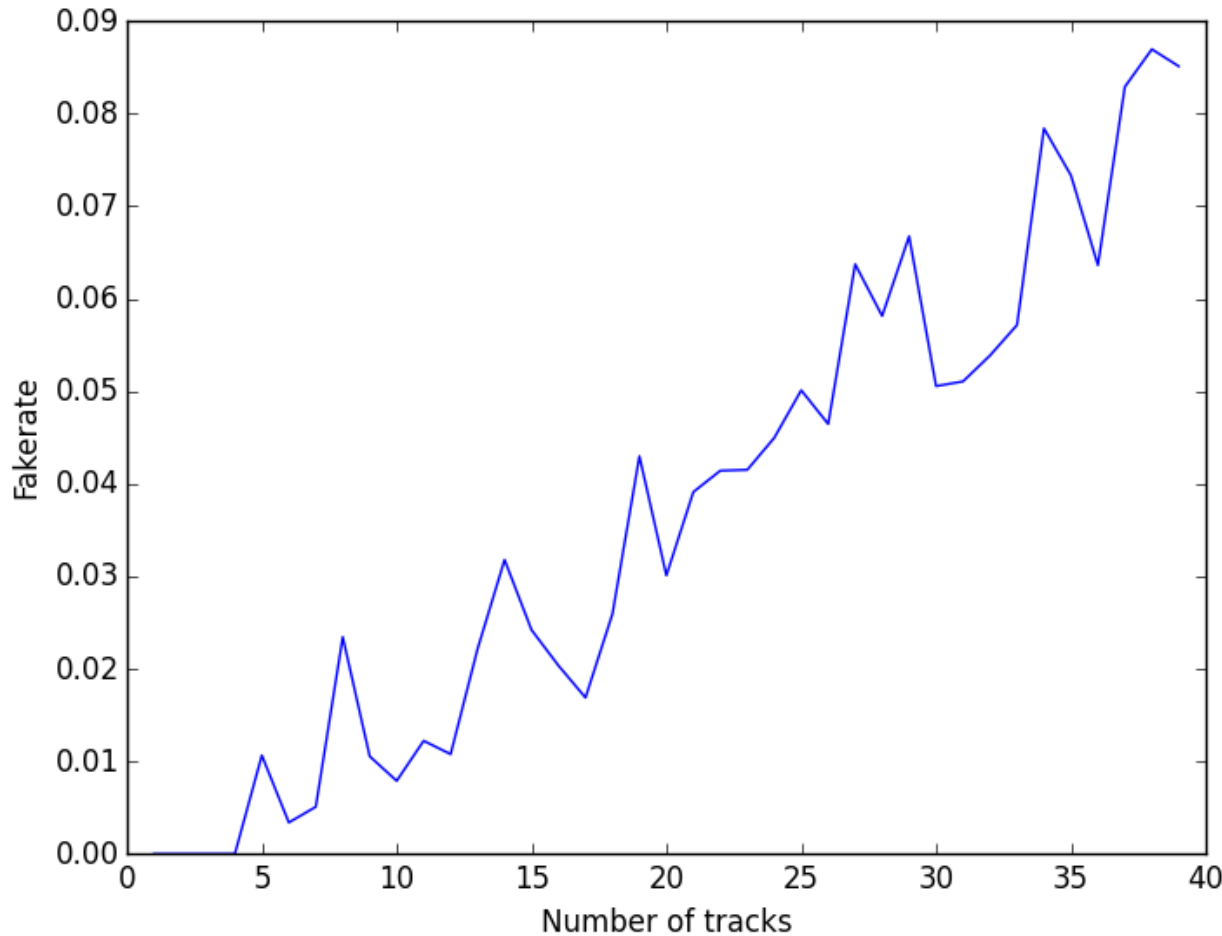


Gemittelt über
200 Durchläufe

$$\alpha = 2\pi/40 = 0.157$$

Wie erwartet ist
Efficiency
unabhängig
von Anzahl an
Tracks

Anzahl Tracks → Fakerate



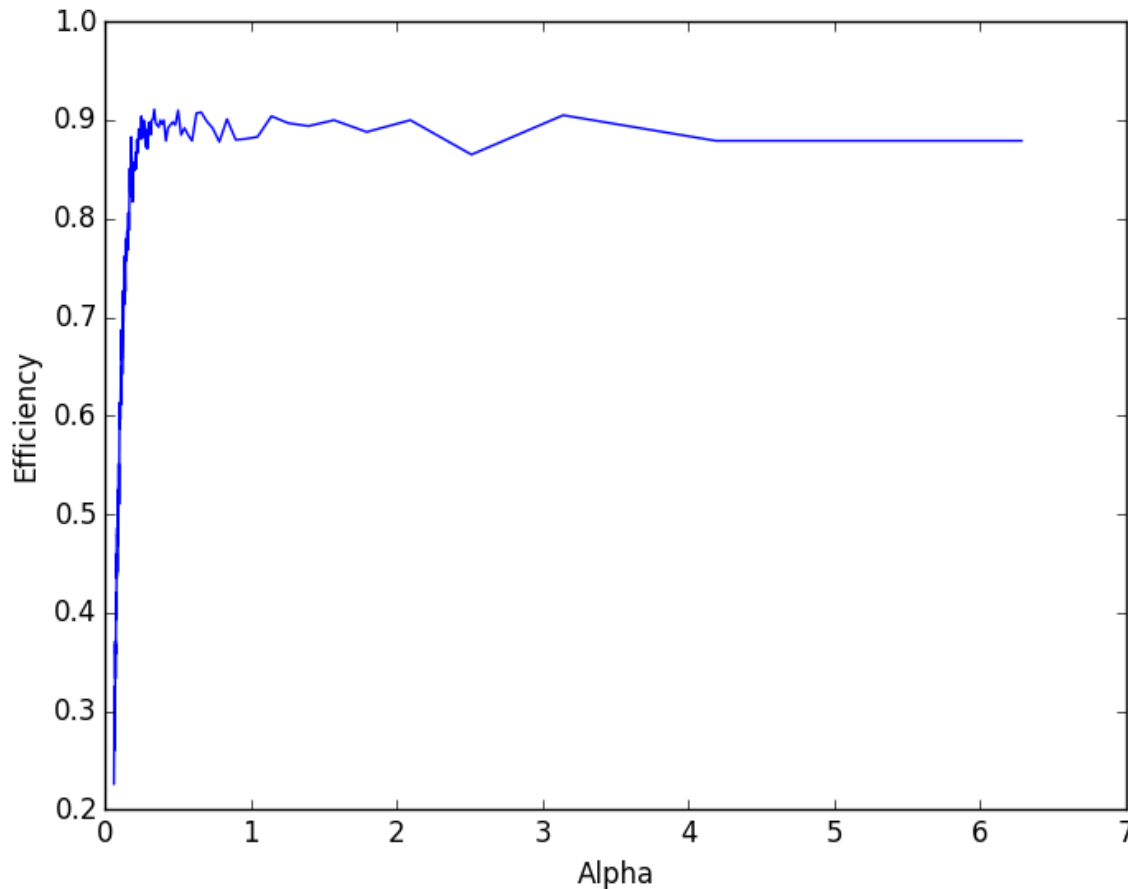
100 Durchläufe

20 Tracks

$\alpha = 0,157$

Nicht eindeutig,
um was für eine
Funktion es sich
handelt

$\alpha \rightarrow$ Efficiency



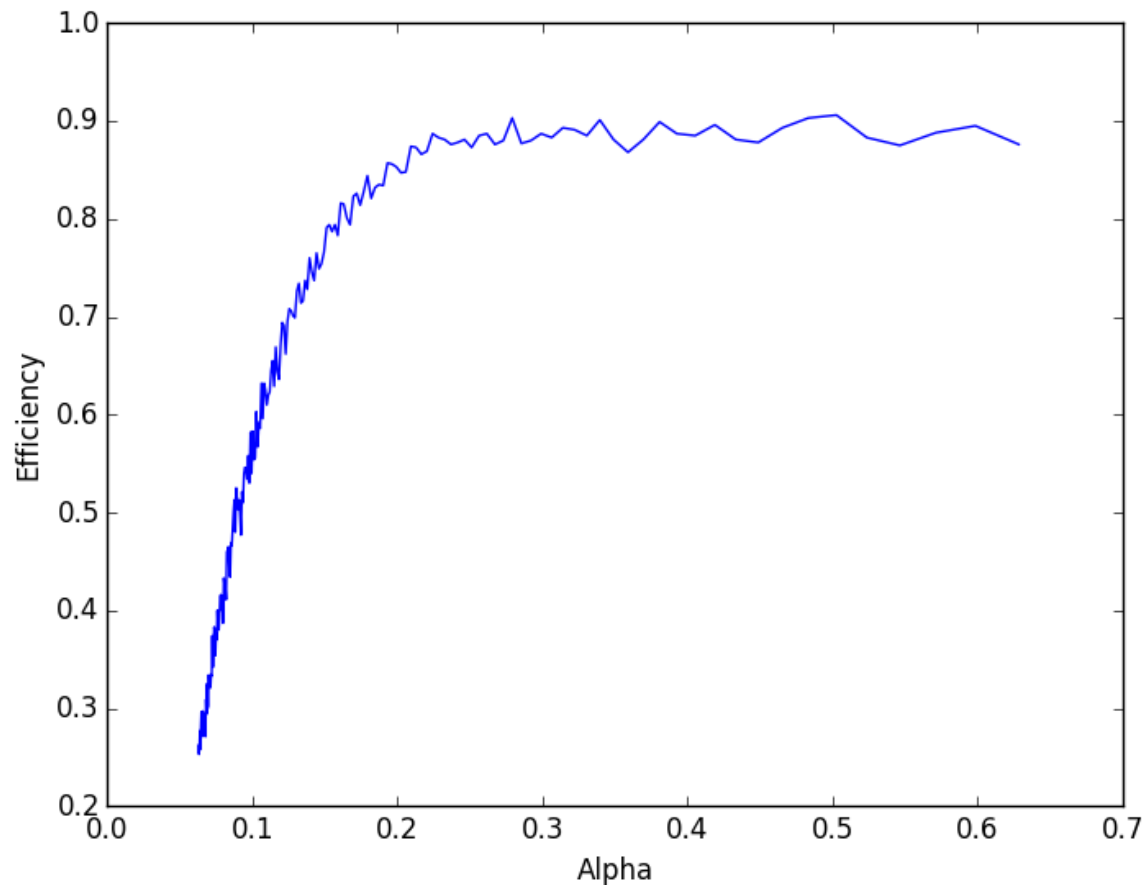
1000 Durchläufe

1 Track

Kaum Veränderung
ab $\alpha = 0.3$

Nur 1 Track zu
verwenden legitim,
da Unabhängigkeit
der Efficiency
gezeigt

$\alpha \rightarrow$ Efficiency



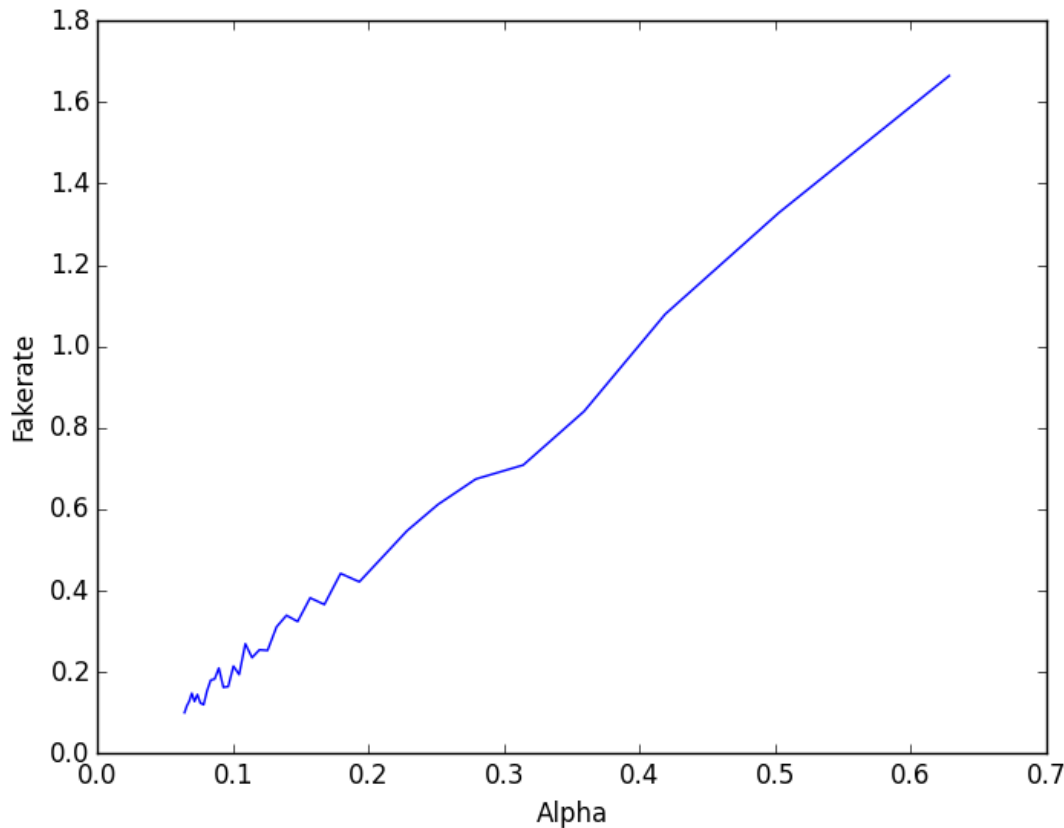
1000 Durchläufe

1 Track

Kaum Veränderung
ab $\alpha = 0.3$

Nur 1 Track zu
verwenden legitim,
da Unabhängigkeit
der Efficiency
gezeigt

$\alpha \rightarrow$ Fakerate



1000 Durchläufe

20 Tracks

5er-Schritte

Scheinbare

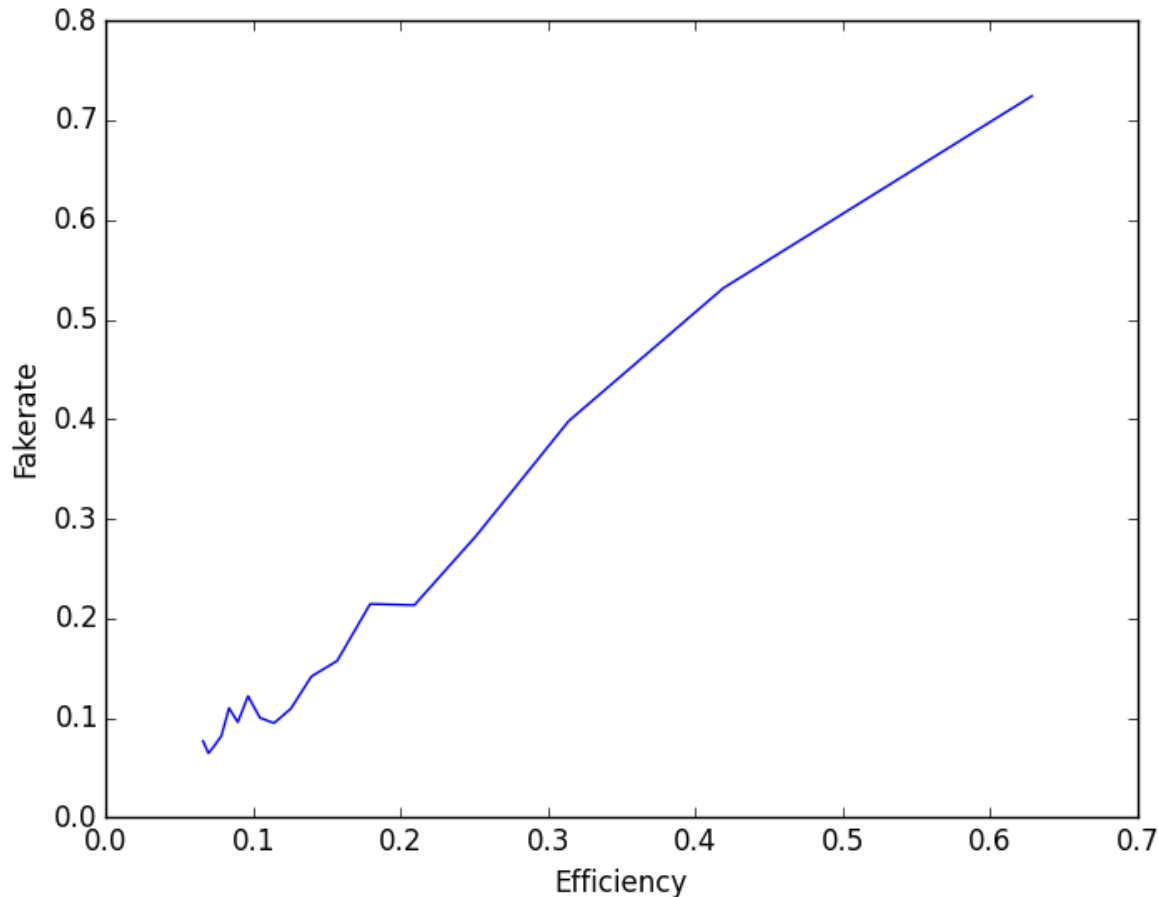
Linearität

vermutlich wegen

geringer

Trackanzahl

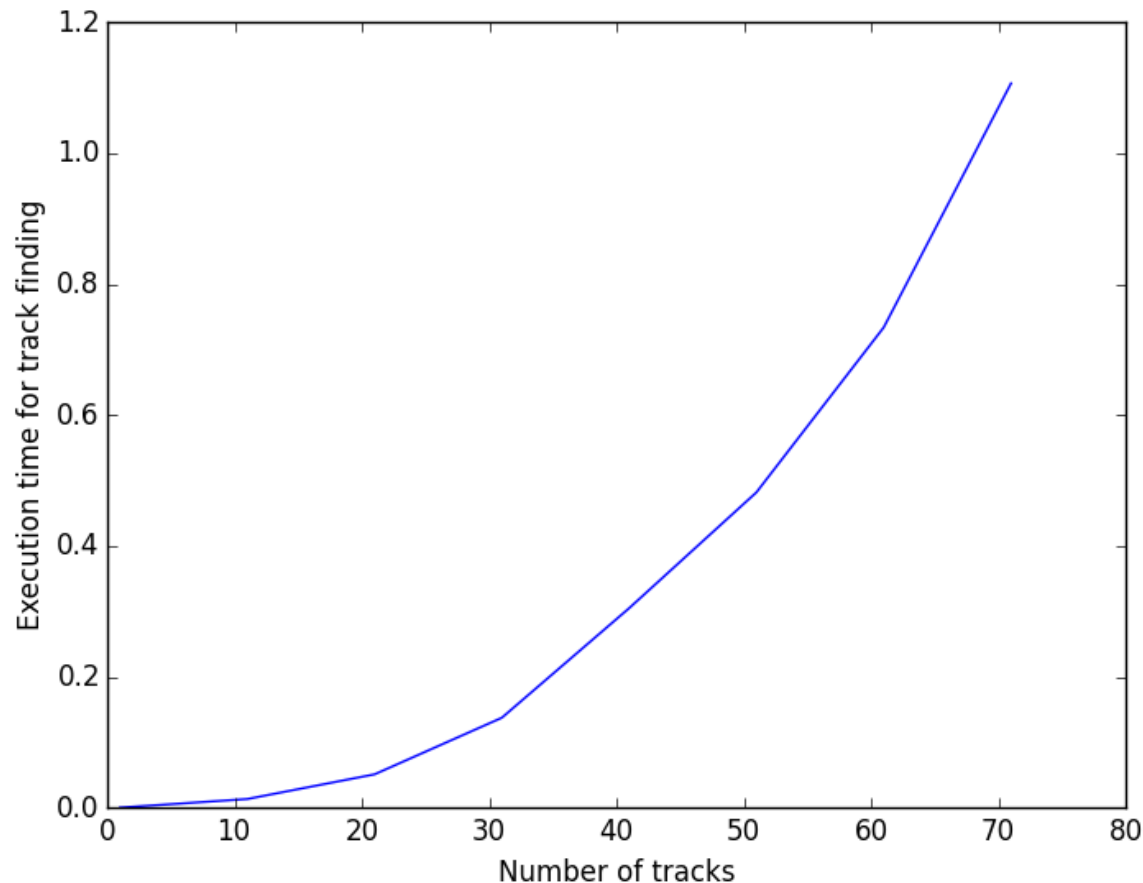
Efficiency \rightarrow Fakerate (ändere α)



Wie zu erwarten
wird höhere
Efficiency mit
höherer Fakerate
erkauft

Dass Funktion
linear aussieht
wieder auf geringe
Trackzahl
zurückzuführen

Anzahl Tracks → Ausführungszeit



200 Durchläufe

10er-Schritte

$\alpha = 0,157$

Exponentielles
Wachstum

→ Algorithmus
für große
Teilchenzahlen
problematisch

Weiteres Laufzeitproblem

- Haben 2 Teilchen sehr ähnliche Parameter, so würden 2^N Tracks gefunden, wobei $N :=$ Anzahl an Detektorzylindern
- Das wären z. B. bei 40 Zylindern ca. 10^{12}
- Problem z. B. lösbar durch Bildung von Clustern

Weiterführende Ansätze

- Vorstellung der Hits als Knoten in Graph: Berechne Gewichte für Kanten
 - u. A. basierend auf den hier verwendeten Kriterien (Winkeldifferenz, Steigungsunterschied)
 - Nehme Kanten nur in Pfad auf, wenn Gewicht größer als Schwelle
 - Beziehe durchschnittliches Kantengewicht auf bisherigem Pfad ein
- Begrenze worst-case Laufzeit, indem nur bestimmte Anzahl an Pfaden verfolgt wird
- Benutze zur Berechnung der Gewichte maschinelles Lernverfahren, z. B. Logistische Regression → Nähere Ableitungen durch Differenzenquotienten; Nimm gemittelte Werte aus vielen Durchläufen

Vielen Dank für Ihre Aufmerksamkeit.