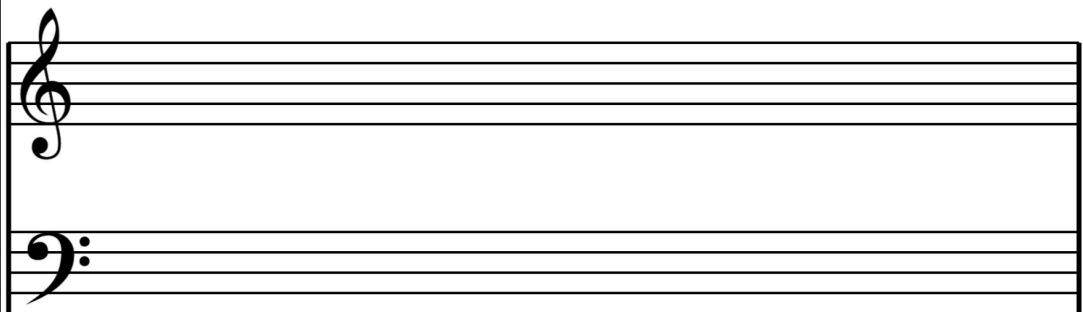
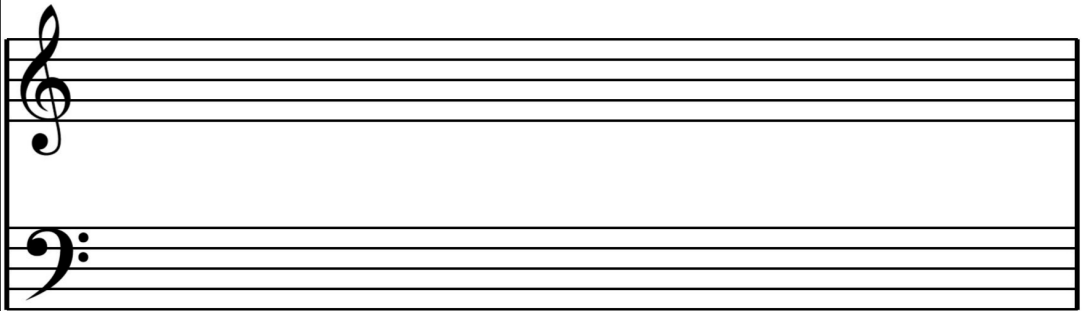


Einsatz der Editierdistanz für einen Musikerkennungsalgorithmus



```
def partialDB(database, directory, intrack, n, m, mpQueue):
    #list of all tracks (already transformed to delta lists)
    #format = (filepath and track name, [d1, d2, ..., d(n-1), d(n)]) , d1 being the distance between note and
    tracks = []

    #Separation of different processes
    for k in range(int(m * (len(database)/n)), int((m+1) * (len(database)/n))):
        filename = database[k]
        if filename.endswith(".mid"):
            #creates the exact filepath of the current file (filename)
            fp = directory + "/" + filename
            absolute = MidiId.getTracks(fp)
            deltas = MidiId.findDeltas(absolute, fp)
            tracks.extend(deltas)

    #makes a delta list from the input track
    track = MidiId.getTrack(intrack, 1)
    inDeltas = MidiId.findDeltas(track)

    #calculates the distances between all tracks in the tracks list and the deltas list
    #the number indicates how close the input track is to the DB tracks (smaller number means less different)
    mpQueue.put((id1.distanceAll(tracks, inDeltas, 1, 1, 1)))

def main():
    #number of Processes used
    n = 10
    #list of Process Objects
    Processes = []
    for i in range(n):
        Processes.append(0)
```

Mattia Brand, M22a
Betreut durch Stefan Rothe

