

# 썸여림 유사도 알고리즘

 Minseo's algorithm(Dynamic Similarity).ipynb

[https://github.com/dalabdgw/Research\\_and\\_Experimental\\_Results/tree/main/Piano%20Performance%20Evaluation/MinseoPark/compare\\_algorithm](https://github.com/dalabdgw/Research_and_Experimental_Results/tree/main/Piano%20Performance%20Evaluation/MinseoPark/compare_algorithm)

기존 데이터로 평가시

```
... ..  
[2]: 썸여림 유사도 : 7.35%
```

새로운 데이터로 평가시

---

```
In [16]: dynamic_similarity = calculate_similarity_dynamic(input_file, target_file)  
print(f'썸여림 유사도 : {dynamic_similarity}')
```

```
썸여림 유사도 : 35.87%
```

# 셈여림 변화 일관성 비교 알고리즘

 Minseo's algorithm(Dynamic Similarity Consistency).ipynb

[https://github.com/dalabdgw/Research\\_and\\_Experimental\\_Results/tree/main/Piano%20Performance%20Evaluation/MinseoPark/compare\\_algorithm](https://github.com/dalabdgw/Research_and_Experimental_Results/tree/main/Piano%20Performance%20Evaluation/MinseoPark/compare_algorithm)

기존 데이터로 평가시

```
[2]: dynamic_consistency = 1.0000
[3]: 셈여림 변화 일관성 : 15.52%
[4]: 새로운 데이터로 평가시
```

새로운 데이터로 평가시

```
: dynamic_change_score = calculate_dynamic_change_consistency(input_file, target_file)
print(f'셈여림 변화 일관성 : {dynamic_change_score}')
```

셈여림 변화 일관성 : 32.26%

	sec	msg_type	channel	note	velocity	new_dynamic	dynamic	accent	count	main_vol	depth	pedal	pan
0	0.0					ppp	ppp	0	0	100	0	0	0
1	0.1	['note_on', 'note_on']	[0, 0]	[60, 60]	[21, 21]	ppp	ppp	0	2	0	0	0	0
2	0.2					ppp	NaN	0	0	0	0	0	0
3	0.3	['note_on', 'note_on']	[0, 0]	[64, 64]	[32, 32]	ppp	ppp	0	2	0	0	0	0
4	0.4					ppp	NaN	0	0	0	0	0	0

	sec	msg_type	channel	note	velocity	new_dynamic	dynamic	accent	count	main_vol	depth	pedal	pan
0	0.0	['note_on']	[0]	[60]	[61]	p	p	0	1	[100, 100, 100]	[127, 127]	127	[64, 64]
1	0.1					p	NaN	0	0	0	0	0	0
2	0.2	['note_on']	[0]	[64]	[57]	p	p	0	1	0	0	0	0
3	0.3					p	NaN	0	0	0	0	0	0
4	0.4	['note_on']	[0]	[67]	[56]	p	p	0	1	0	0	0	0

빨간 부분 데이터가 서로 다른 곳이 많아서, 현재 비교 알고리즘의 성능이 안 좋게 나오는 것이 정  
 확한 것으로 추측됩니다. 따라서  
 정확한 쉼여림 테스트 데이터를 제작해서 비교 알고리즘을 테스트 해보아야 할 것 같습니다!