Technical Report

Developing Loudness Logging System

Duckhyung Ryu

Seoul Broadcasting System

MAY 2024

SBS

Table of Contents

Introduction

Background

지상파 방송 Loudness War로 인한 오디오 퀄리티 저하를 방지하기 위한 2018년 7월 시행된 과학기술정보통신부의 시행령 [KO-07.0114 디지털 방송 음성 레벨 운용 기준]에 따라, 대한민국에서 운용되는 방송국 프로그램은 Loudness가 -24ILFKS 로 제한되며 운용상의 ±2dB이내 오차만 허용된다. 또한 방송국은 송신 Headend에서 일일 편성을 기준으로 Loudness를 측정하고 기록해 이를 일정 주기별로 방송통신위원회에 보고할 의무를 가지게 되었다.

당사는 해당 시행법령 시행부터 Mediaproxy사의 Loudness Logger 시스템을 도입해 사용해 왔고, 수천만원의 예산을 들여 해당 장비를 운용하고 있다. 해당 Logger는 현재 DSK와 Caption Inserter의 사이 구간에서 실시간 Loudness를 컨트롤하는 Loudness Controller장비의 정보를 받아 실시간 Loudness를 기록해두고 있고 편성정보와 함께 프로그램별 Integrated Loudness를 측정하고 있다. 본인은 Integrated Loudness에 대한 측정을 직접 할 수 있는 방법을 연구해 봄으로써 보다 더 정확하고 유연한 시스템을 개발하게 되었다.

Expectation

본인은 이번 연구를 통해 3가지의 기대점을 가지고있다.

- 정확한 프로그램별 Loudness(LKFS, LUFS) 측정을 제시함으로써 과학기술정보통신부와 ITU-R 국제표준에 더 적합한 Loudness 측정을 통해 당사의 기술품질을 높인다.
- 둘째, 당사의 자체 기술력을 높임으로써 향후에 추가 요구사항이나 요청사항에 대한 대응 유연성을 높이고 추후 관련 구매에 대한 협상력을 높인다.
- 셋째, 현재 시스템에 소비되고 있는 예산을 절약하게 함으로써 당사가 직면한 지상파 방송국 위기 극복에 도움을 준다.

Design Proposal

Loudness Measurement

Loudness는 절대적인 파워를 dB로 나타내는 것이 아닌 인간이 주관적으로 느끼는 음량의 정도를 나타내는 것으로 Equal-Loudness Curve를 기준으로 기본 음성에 필터를 씌워 측정을 하는것을 기본으로 한다. 이를 나타내는 용어로 LKFS (Loudness K-weighted Full Scale) 을 사용하며 대한민국의 과학기술정보통신부에서는 ITU-R BS.1770의 표준을 따르고 있다. 참고로 일반적으로 EBU에서 사용하는 용어인 LUFS라는 용어가 더 널리 사용되는데 이 둘을 완전히 같은 개념이다.

Loudness는 4개의 단계를 거쳐서 계산된다.

1. K frequency weighting
2. 각 채널별 Mean square calculation

3. 채널 가중치 필터 적용 및 합산

4. 400ms 단위 게이팅(Gating)

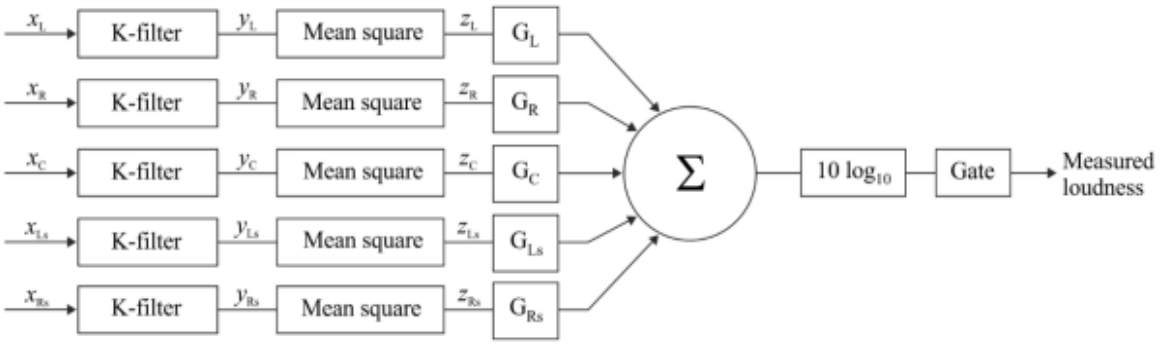


Figure 1: Block diagram of multichannel loudness algorithm (ITU BS.1770)

K-filter weighting는 일반적으로 저저역대에 둔감한 인간의 청각 시스템을 반영하는 작업으로, 여러분야에서 계속해서 연구되고 있으며 사용되는 분야마다 다른 형태가 있다. 대한민국 방송시스템에서는 BS.1770에 제시된 shelving filter와 high-pass filter 두 단계로 구성된 K-filter를 기준으로 LKFS를 구한다. 또한 Channel weighting은 Left, Right, Centre, Left surround, Right surround 이 각각 [1.0, 1.0, 1.0, 1.41, 1.41] 로 적용한다.

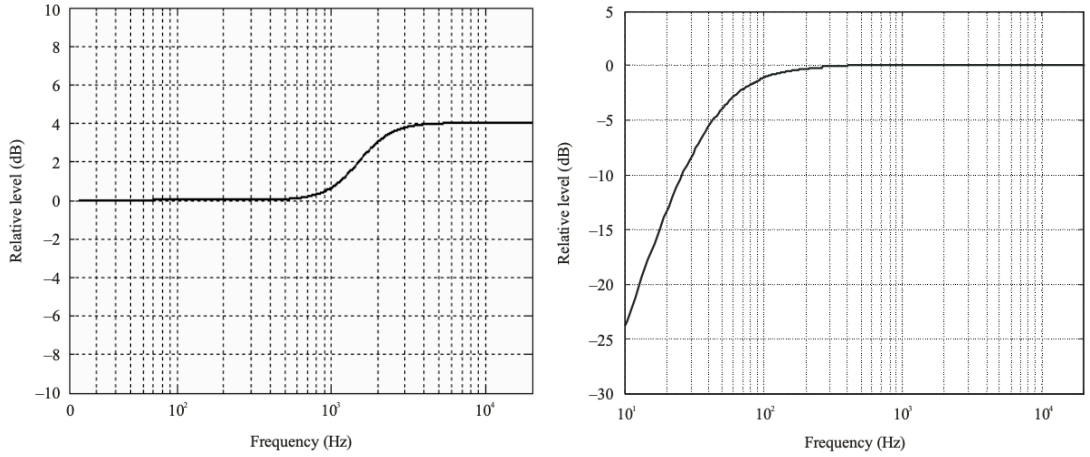


Figure 2. Filter response of first and second stage (ITU BS.1770)

Gating의 경우 BS.1770-3에서 추가된 부분인데, mute 혹은 small ambient sound만 있는 부분들이 LUFS값에 영향을 주지 않게 하기 위해 도입되었다. 소리가 있는부분은 음량을 엄청 크게 해버리고 나머지 silence한 구간들을 둬서 평균을 내려버린다면 Loudness 규제 도입 취지에 맞지 않기 때문이다. 따라서 400ms 단위의 Momentary Loudness를 구하고 우선 absolute threshold에 해당하는 -70dB이하의 값들은 버린 후 전체 LKFS값을 구한다. 그리고 relative threshold라 불리는 전체 LKFS값에서 -10dB 이하인 Momentary Block들은 또 제거함으로써 진짜 소리가 제대로 나오는 부분만을 걸러낸다.

Loudness Controller와 같이 실시간 Loudness측정을 하는 시스템들은 매우 짧은시간동안의 Loudness를 측정해서 나타내는 것으로 보통 400ms 시간 범위를 가지며 75%의 Overlay를 거치는 Momentary Loudness를 측정하는 것으로 음향 신호의 빠른 변화를 감지하는데에 사용된다. 현재 당사에서 사용하는 Loudness logger 시스템은 이렇게 Controller 단에서 실시간으로 계산된 Loudness값들을 시간대별로 저장해 두었다가 편성표를 기준으로 위의 게이팅 작업과 계산을 마쳐 프로그램의 Integrated Loudness를 구하는 식으로 보인다.

그러나 Loudness Controller가 아닌 Logger는 사실 실시간 LKFS를 알아야 할 필요가 없고, 전일 편성표가 주어졌을 때 해당 프로그램들에 대해서만 정확한 Loudness 계산을 해주면 되는 시스템이기에, 실시간으로 빠른 처리를 하는 것 보다는 프로그램 전체 오디오데이터를 통한 정확한 측정 및 로깅을 해도 되겠다고 판단하였다.

Python 프로그래밍 언어로 시스템을 개발하기 위해 LUFS계산은 ITU-R BS.1770을 준수하고 있는 오픈소스 라이브러리 Pyloudnorm (https://github.com/csteinmetz1/pyloudnorm)를 이용하였다. LUFS계산은 K-weighted filtering, gating, 계산중 반올림 등의 과정에서 같은 알고리즘을 따르더라도 구현방식마다 조금씩 결과값의 차이가 생기는데, Pyloudnorm(default)의 경우 BS.2217 Measurement Meter 데이터에서 Target Loudness에 매우 근접하며 tolerance for compliance인 0.1dB 이상의 오차가 생기지 않은 ITU-R에 적합한 LKFS구현이기에 해당 라이브러리를 사용하였다.

File	Target	Implementation							
		pyloudnorm		loudness.py	ffmpeg	libebur128	Essentia	Audition	youlean
		Default	De Man						
FrequencySweep	-18.0	-18.03	-17.99	-17.99	-18.00	-18.00	-18.18	-18.03	-18.02
25Hz_2ch	-23.0	-23.00	-22.99	-22.99	-23.10	-23.00	**-26.37**	-23.04	-23.02
100Hz_2ch	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	**-22.86**	-23.04	-23.02
500Hz_2ch	-23.0	-23.04	-22.99	-22.99	-23.10	-23.00	-22.99	-23.04	-23.02
1000Hz_2ch	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	-23.00	-23.04	-23.02
2000Hz_2ch	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	-23.00	-23.04	-23.02
10000Hz_2ch	-23.0	-23.04	-22.99	-22.99	-23.10	-23.00	-23.00	-23.04	-23.02
25Hz_2ch	-24.0	-24.00	-23.99	-23.99	-24.10	-24.00	**-27.21**	-24.04	-24.02
100Hz_2ch	-24.0	-24.03	-23.99	-23.99	-24.10	-24.00	-23.92	-24.04	-24.02
500Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-23.99	-24.04	-24.02
1000Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-24.00	-24.04	-24.02
2000Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-24.00	-24.04	-24.02
10000Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-24.00	-24.04	-24.02
RelGateTest	-10.0	-10.07	-10.03	-10.03	**-9.60**	-10.00	-10.03	-10.07	**-10.15**
AbsGateTest	-69.5	-69.49	-69.45	**-71.46**	-69.50	-69.50	-69.45	-69.49	-69.55
Mono_Voice+Music	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	-22.97	-23.03	-22.98
Mono_Voice+Music	-24.0	-24.03	-23.99	-23.99	-24.10	-24.00	-23.97	-24.04	-23.98
Stereo_VinL+R	-23.0	-23.03	-22.98	-22.98	-23.10	-23.00	-22.97	-23.02	-22.99
Stereo_VinL+R	-24.0	-24.02	-23.98	-23.98	-24.10	-24.00	-23.97	-24.02	-23.98

Table 1. Comparison of loudness algorithm implementation (Ref.[5])

System Outline

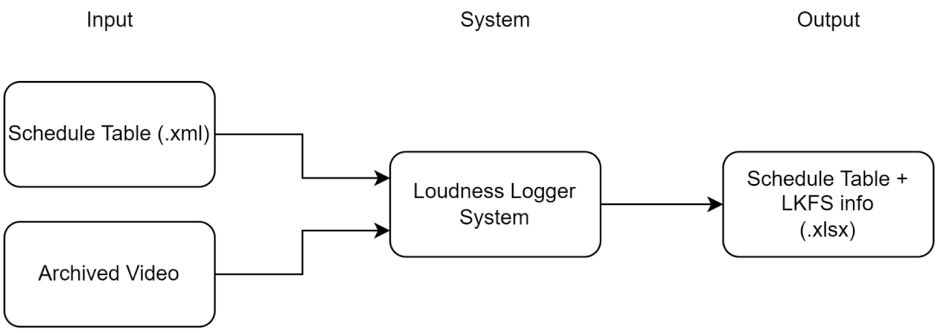


Figure 3. System Outline Block Diagram

시스템은 Python을 통해 개발했으며 편성표를 넣으면 해당 프로그램들을 아카이브된 영상자료에서 찾아서 LUFS계산을 하고 제출 가능한 형태의 엑셀 테이블을 리턴하는 방식으로 설계하였다.

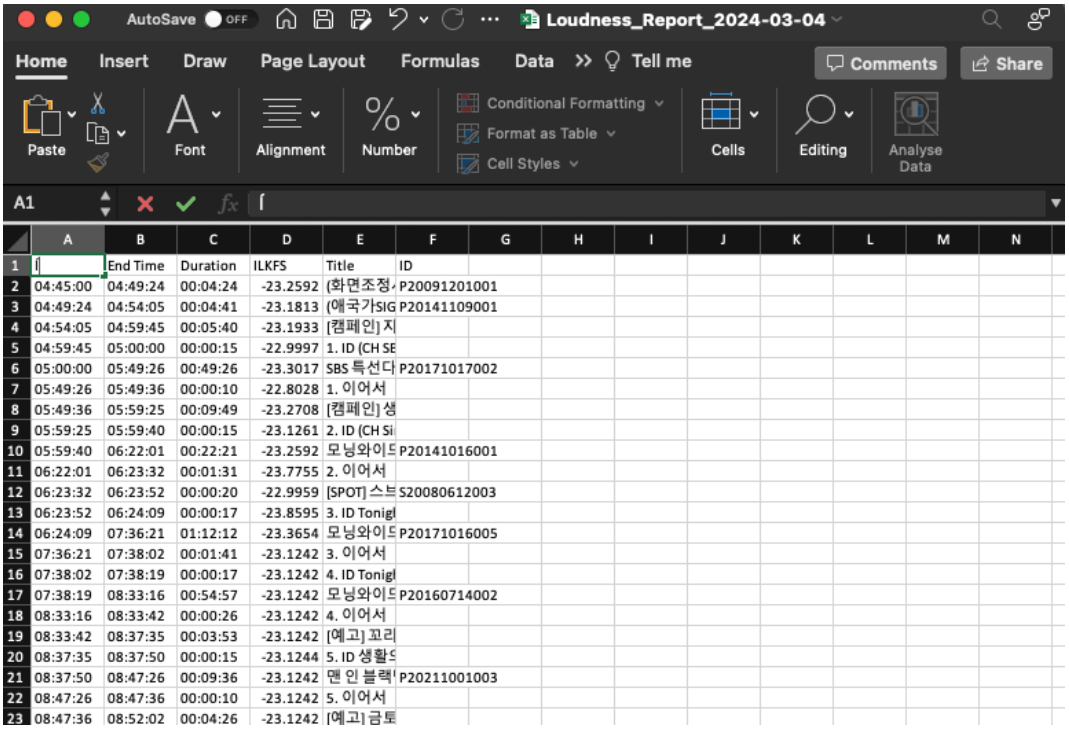
Input단에 해당하는 편성표는, 기존 Mediaproxy 장비에 Input으로 들어가던 편성정보와 동일한 xml형태의 정보를 활용하였다. 해당 편성표는 일일단위로 하지만 Technical한 하루 단위가 아니라 편성상의 하루 기준이기에, 대략 오전 4시~5시 사이에 시작해서 다음날 정파 전까지의 정보들이 기록되어 있다. 여기엔 OnAirDate, StartTime, Duration, PGMID, EventTitle정보가 있는데, PGM ID의 경우 진짜 프로그램이 아닌 경우 (캠페인, 예고, 광고 등) 값이 없다. 그리고 과기정통부의 '텔레비전 방송프로그램 표준 음량기준 준수 안내문'의 서면 제출자료에 형식을 맞추기 위해 Output 엑셀 테이블에는 프로그램이 끝나는 시각인 End Time을 Start Time과 Duration을 바탕으로 계산해서 해당 정보를 추가하였다.

Archived Video는 기존 서버실에서 PGM원본을 1시간단위로 15일치 저장하고 있는 장비에 접속하고, 편성시간과 타임코드를 정확히 맞추기 위해 당일 00시 부터 다음날 까지의 비디오 데이터를 모두 가져온다. Python subprocess로 ffmpeg을 사용해서 하나로 쭉 이어진 wav파일로 트랜스코딩을 한 후 편성표대로 ffmpeg을 이용해 이 원본 wav파일을 타임코드에 맞게 자르고 순차적으로 Integrated LUFS를 계산하도록 했다.

즉 기존 Loudness Controller상의 Momentary Loudness를 사용하는게 아니라 프로그램 전체 시간에 대한 통합(Integrated) Loudness를 정확히 구함으로써 과기정통부와 BS.1770 에서 정의하는 프로그램 Loudness를 정확히 측정하는 것이다.

Python자체에도 비디오나 오디오를 concat하고 컷팅할 수 있는 라이브러리가 있지만 메모리와 CPU, GPU등의 사용량을 봤을 때 긴 wav파일을 로드하려는 순간 이를 메모리에 프로세스로 올리기 때문에 잘못해서 메모리 용량이 부족하면 컴퓨터 전체 프로세스가 killed 될 수 있었다. 그러나 subprocess로 ffmpeg을 사용할 때에는 컴퓨팅자원을 안정적으로 활용할 수 있어 해당 방식을 사용하게 되었다.

Result



	A	B	C	D	E	F
1	[	End Time	Duration	ILKFS	Title	ID
2	04:45:00	04:49:24	00:04:24	-23.2592	(화면조정	P20091201001
3	04:49:24	04:54:05	00:04:41	-23.1813	(애국가SIG	P20141109001
4	04:54:05	04:59:45	00:05:40	-23.1933	[캠페인] 지	
5	04:59:45	05:00:00	00:00:15	-22.9997	1. ID (CH SE	
6	05:00:00	05:49:26	00:49:26	-23.3017	SBS 특선다	P20171017002
7	05:49:26	05:49:36	00:00:10	-22.8028	1. 이어서	
8	05:49:36	05:59:25	00:09:49	-23.2708	[캠페인] 생	
9	05:59:25	05:59:40	00:00:15	-23.1261	2. ID (CH Si	
10	05:59:40	06:22:01	00:22:21	-23.2592	모닝와이드	P20141016001
11	06:22:01	06:23:32	00:01:31	-23.7755	2. 이어서	
12	06:23:32	06:23:52	00:00:20	-22.9959	[SPOT] 스브	S20080612003
13	06:23:52	06:24:09	00:00:17	-23.8595	3. ID Tonigl	
14	06:24:09	07:36:21	01:12:12	-23.3654	모닝와이드	P20171016005
15	07:36:21	07:38:02	00:01:41	-23.1242	3. 이어서	
16	07:38:02	07:38:19	00:00:17	-23.1242	4. ID Tonigl	
17	07:38:19	08:33:16	00:54:57	-23.1242	모닝와이드	P20160714002
18	08:33:16	08:33:42	00:00:26	-23.1242	4. 이어서	
19	08:33:42	08:37:35	00:03:53	-23.1242	[예고] 꼬리	
20	08:37:35	08:37:50	00:00:15	-23.1244	5. ID 생활	
21	08:37:50	08:47:26	00:09:36	-23.1242	맨 인 블랙	P20211001003
22	08:47:26	08:47:36	00:00:10	-23.1242	5. 이어서	
23	08:47:36	08:52:02	00:04:26	-23.1242	[예고] 금토	

Figure 4. Program Output for data of 4 MAR

편성 Input에 대해 안정적으로 ILKFS가 계산된 과기정통부의 서면 제출자료 형식에 맞춘 Excel 파일이 출력되는 것을 볼 수 있고, 계산된 LKFS값들을 기존 장비에서 나온 값들과 비교해 봤을 때 평균 0.14dB, 최대 0.44dB의 차이를 보여준다. [A1]

avg diff	0.144196266
max diff	0.447114286

Table 2. LKFS Value Difference Between ASIS, TOBE System

BS.1770을 Implementation한 오픈소스 라이브러리를 사용했고 정확한 프로세스를 통해 프로그램 Integrated LKFS를 측정했다. 하지만 앞서 말했듯 LKFS계산은 알고리즘 구현방식에 따라 값의 미세한 차이가 생기기 때문에 이는 무엇이 맞고 틀리다 라고 하기 보다는 서로 용인 가능한 정도의 차이가 있다 라고 볼 수 있으며 Pyloudnorm 라이브러리는 BS.2217 Measurement Meter에서 Target Loudness에 0.1dB 이상의 오차가 없었던 점을 감안할 때 더더욱 해당 LKFS의 값이 정상적임을 알 수 있다.

References:

[1] 과학기술정보통신부, “디지털 텔레비전 방송프로그램 음량 등에 관한 기준”, 과학기술정보통신부고시 제2018-39호, 2018. 7. 3, Accessed May. 3, 2024 [Online] Available: <https://law.go.kr/LSW/admRulLsInfoP.do?admRulSeq=2100000134169>

[2] ITU-R, “Algorithms to measure audio programme loudness and true-peak audio level”, Recommendation BS.1770-3 (08/2012), Accessed May. 3, 2024 [Online] Available: <https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-3-201208-S!!PDF-E.pdf>

[3] 조용성 최동준, “디지털 방송 음량 레벨 운용 기준”, ETRI저널

[4] KT스카이라이프 방송운용팀, “오디오 라우드니스”, 방송과기술 2016년 10월호, Available: <http://tech.kobeta.com/wp-content/uploads/2016/10/23318.pdf>

[5] Christian J. Steinmetz, Joshua D. Reiss, pyloudnorm: A simple yet flexible loudness meter in Python, Available: <https://csteinmetz1.github.io/pyloudnorm-eval/paper/pyloudnorm_preprint.pdf>

[6] ITU-R, “Compliance material for Recommendation ITU-R BS.1770”, BS.12217-2, Accessed May. 9, 2024 [Online] Available: <https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-BS.2217-2-2016-PDF-E.pdf>

Appendix:

A1: 기존 시스템과의 결과 비교 (2024년 3월 24일 데이터)

TO-BE	AS-IS	Difference
-23.2758	-23.134	0.141804582
-23.1624	-23.043	0.119436572
-23.198	-23.071	0.127011871
-23.6395	-23.537	0.102528488
-23.4169	-23.324	0.092881669
-23.4832	-23.258	0.225170376
-23.2623	-23.15	0.112284733
-22.6642	-22.45	0.214206078
-23.0803	-22.995	0.085263279
-23.1645	-22.993	0.171537522
-22.838	-22.661	0.17701612
-24.121	-23.919	0.202013903
-23.2033	-23.112	0.091309335
-23.2945	-23.135	0.159467975
-23.8097	-23.625	0.184683227
-23.5165	-23.404	0.112535217
-23.2936	-23.206	0.087614032
-23.2395	-23.1	0.139495107
-22.8968	-22.764	0.132796338
-23.1985	-23.075	0.123490208
-23.6073	-23.464	0.143323739
-23.1743	-23.035	0.139261931
-23.0939	-22.87	0.223921404
-23.1304	-23.037	0.093371622
-23.2133	-23.106	0.107284389
-23.1702	-23.038	0.132227277
-22.5927	-22.314	0.278681957
-23.0516	-22.975	0.076591061
-23.4905	-23.368	0.122514475

-23.1927	-23.078	0.114676345
-23.4822	-23.153	0.3291681
-23.2406	-23.101	0.139598337
-22.9976	-22.849	0.148561942
-23.2386	-23.105	0.133622261
-22.7856	-22.742	0.043597522
-23.1685	-23.075	0.09349275
-23.0428	-22.931	0.11177593
-23.2024	-23.085	0.117386002
-23.7176	-23.838	0.120378811
-23.1667	-23.053	0.113726883
-23.1934	-23.09	0.103402025
-23.2026	-23.07	0.132563762
-23.0376	-22.847	0.190565685
-23.0574	-22.979	0.078421489
-23.5418	-23.446	0.095757158
-23.1817	-23.067	0.114699761
-23.1241	-22.677	0.447114286
-23.2462	-23.13	0.116220146
-23.4197	-23.29	0.12972517
-23.1442	-23.009	0.135163851
-23.4803	-23.304	0.176343877
-23.5042	-23.428	0.076241002
-23.147	-23.061	0.085966009
-23.2047	-23.073	0.131713117
-23.0182	-22.848	0.170203118
-23.1645	-23.06	0.104541798
-22.8332	-22.621	0.212173125
-23.1789	-23.038	0.140919992
-23.0547	-22.866	0.188700345
-23.2205	-23.133	0.087548322
-23.7966	-23.648	0.148640108
-23.1716	-23.029	0.142617622
-22.9783	-22.781	0.197343878
-23.2326	-23.087	0.145585258

-23.253	-23.162	0.091005453
-23.1357	-22.968	0.167690059
-23.2357	-23.104	0.13169034
-22.7096	-22.375	0.334551735
-23.2369	-23.09	0.146876593
-23.2629	-23.116	0.146898187
-23.0948	-22.916	0.178826669
-23.1842	-23.142	0.042176512
-23.1758	-23.032	0.143823124
-23.7113	-23.649	0.06230662
-22.8842	-22.798	0.086220541
-23.1254	-22.954	0.171431961
-23.2317	-23.045	0.186707572
-23.2319	-23.216	0.015913123
-23.2269	-23.087	0.139868146
-23.4153	-23.323	0.092310579
-23.1945	-23.106	0.088539658
-23.385	-23.075	0.310014804
-23.2465	-23.13	0.116538122
-23.2457	-23.061	0.184663541
-23.2239	-23.085	0.138908058
-23.3952	-23.257	0.138210248
-22.9872	-22.904	0.083151363
-23.1697	-23.017	0.152741077
-23.1347	-23.014	0.120732111
-23.4789	-23.127	0.351878773
-23.1953	-23.085	0.110347499
-23.2953	-23.182	0.11332832
-22.6634	-22.47	0.193418468
-23.357	-23.224	0.132986351
-22.155	-22.342	0.187009377