A suite of software modules
for advanced video processing workflow

VideoQ Productivity Tools

Training Presentation

September 2025

VQPT

videoq.com

TOC ← *Click on **TOC** n the upper-left corner of any slide for the global **Table Of Contents***

Table of Contents

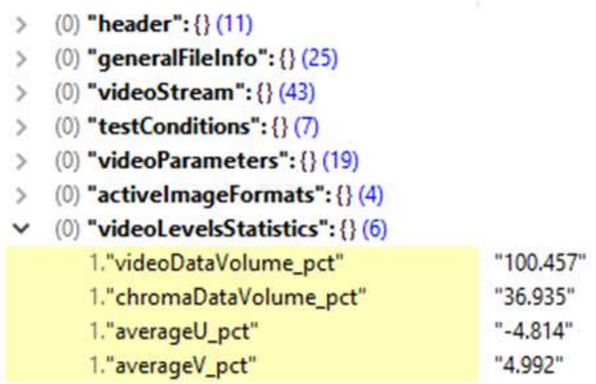
*Click on **PTOC** in the upper-right corner for **VQPT Packages Table Of Contents*** → PTOC

Background Info

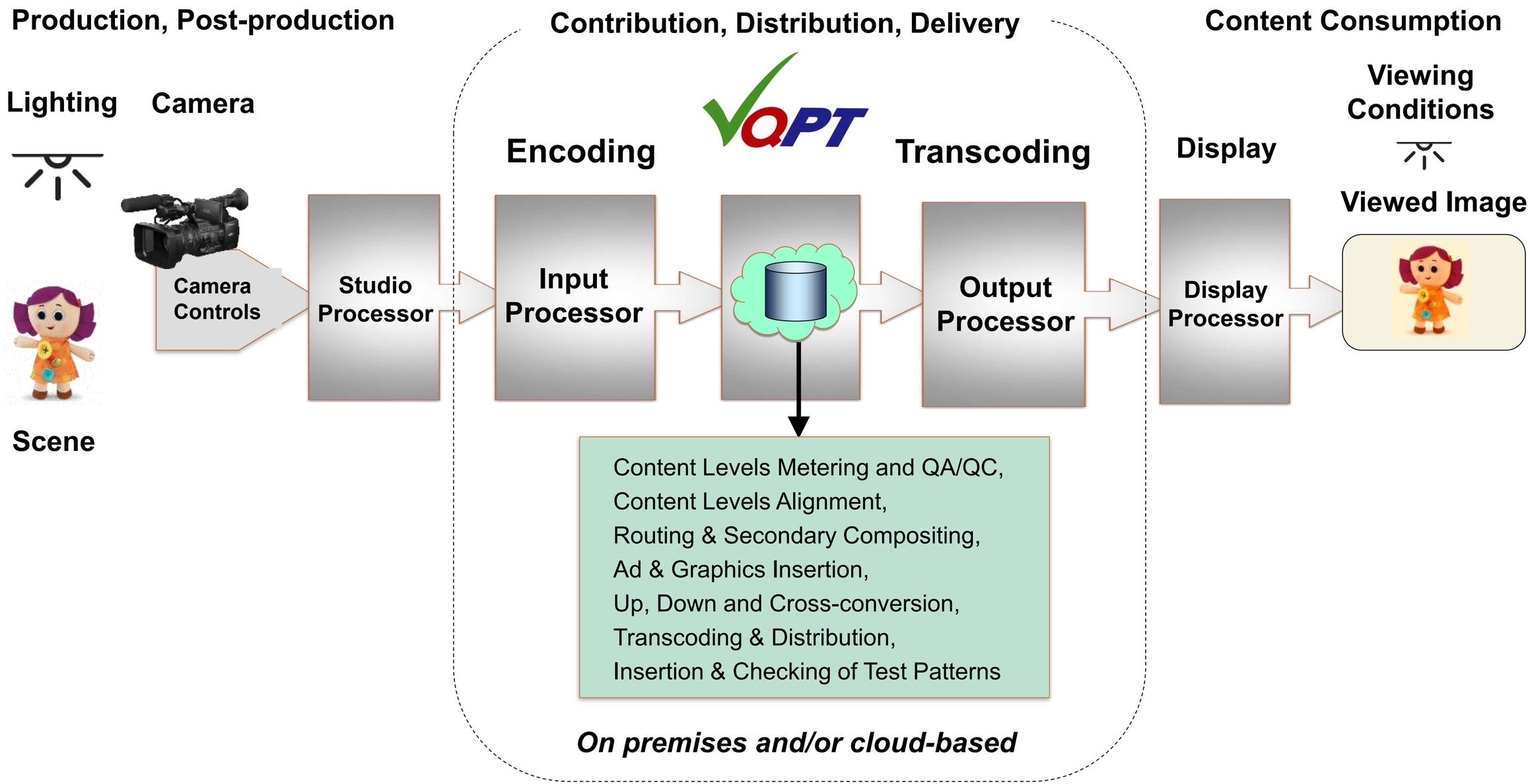
1. VideoQ Productivity Tools are designed “**by engineers, for engineers**”
2. An ever-higher number of channels/programs/titles
3. And a permanently growing number of formats, frames sizes, bitrates, etc.
4. Human resources required for input QC and output QC has escalated
5. A new approach and **new tools** are needed *as demanded by our customers*
6. Hence VideoQ has changed the focus from our traditional T&M tools to

   Automated Productivity Tools

- Automation is essential, but …
- Human intervention cannot be excluded
- Thus, our slogan is: “**Robot-assisted human decisions**”



The Big Picture – Overall System View

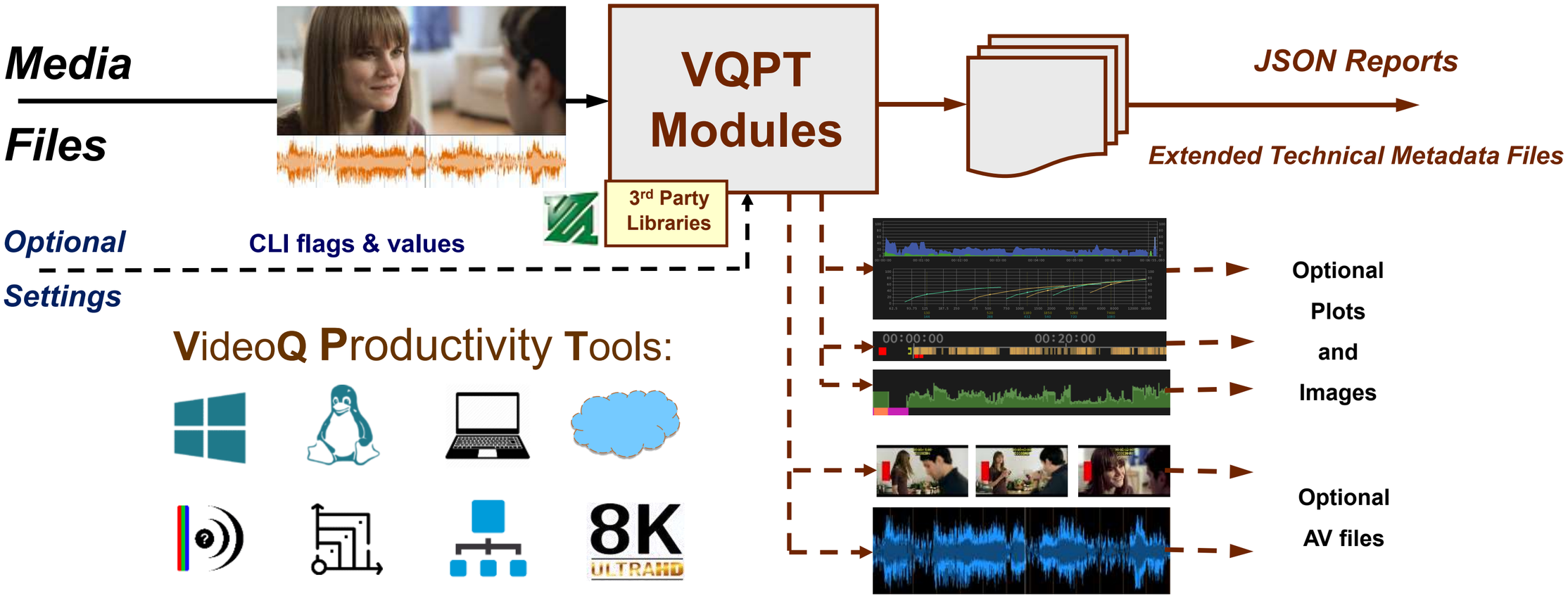


Top Level VQPT Automated Workflow Diagram

VQPT is a suite of portable Windows/Linux CLI programs for on premises and cloud computing.

It can be used for production, post-production and distribution applications.

The program modules can be purchased and used separately or grouped for typical applications.



Principles of Application

With the massive increase of volumes of hardware items and video related software, the strict rules established for the broadcast TV are not always recognized.

The solution is in establishing easy-to-use and straightforward **rules** and matching **tools**.

1. VideoQ **VQPT** is the cloud-based **QA/QC** and **transcoding workflow optimization tool**, that answers this challenge
2. Periodic testing of the **workflow health** should be combined with permanent checking of input and output **AV content parameters**
3. The most efficient methodology of such QA/QC operations is the creation of machine-readable **reports** built by automated program modules and the subsequent review of these reports by a **human operator**
4. Storage of such reports (aka **E**xtended **T**echnical **M**etadata) in the **centralized database**, that allows the **remote access** by authorized users, is vitally important for the efficient **management** of the whole content delivery process

TOC

VQPT Packages

*Click on **PTOC** in the upper-right corner for **VQPT Packages Table Of Contents*** → PTOC

VQPT program modules can be used separately or grouped for the following typical applications:

Pack 1. Target Application: **Workflow Health Tests**

P1TOC

- 1.1 VQMINF – Media File Info Report Generator
- 1.2 VQCBA – VideoQ Color Bars Analyzer, companion program for VQCB test patterns
- 1.3 VQCSA – Compression Stress Analyzer, companion program for VQCST test patterns
- 1.4 VQMA – Video Analyzer for objective video processing chain integrity tests

Pack 2. Target Application: **AV Files Conversion, Encoding and Transcoding**

P2TOC

- 2.1 VQMINF – Media File Info Report Generator
- 2.2 VQBIF – BIF (Base Index Frames) Files Verifier
- 2.3 VQBLA – Bitrate Ladder Analyzer
- 2.4 VQC – HDR-SDR Files Converter
- 2.5 VQCSA – Compression Stress Analyzer, companion program for VQCST test patterns
- 2.6 VQLPN – Audio Loudness Profiler and Normalizer
- 2.7 VQTSF – Transcoding Segments Finder

Pack 3. Target Application: **AV Content Analysis**

P3TOC

- 3.1 VQMINF – Media File Info Report Generator
- 3.2 VQCFA – Captions Files Analyzer
- 3.3 VQFP – Video Frames Profiler
- 3.4 VQLPC – Loudness Profiles Correlator, companion program for VQLPN module
- 3.5 VQLPN – Audio Loudness Profiler and Normalizer
- 3.6 VQPLA – Picture Levels Analyzer

*Click on **P1,2,3TOC** for **Pack 1,2,3 Table Of Contents***

Pack 1: VQPT Modules for Workflow Tests

*Click on **P1TOC** in the upper-right corner for **Pack 1 Table Of Contents***

1.1 VQMINF – Media File Info Report Generator

VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in UTF8 JSON format.

It can read HDR-PQ, HDR-HLG and SDR video as well as LOG video metadata.

VQMINF also calculates unique vqminfEssenceID string and input file MD5 hash string.

1.2 VQCBA – VideoQ Color Bars Analyzer, companion program for VQCB test patterns.

VQCBA reads VQCB test pattern file at particular workflow test point, then checks video and audio levels, detects color space distortions, dynamic range conversion footprints and measures AV Sync errors

1.3 VQCSA – Compression Stress Analyzer, companion program for VQCST test patterns.

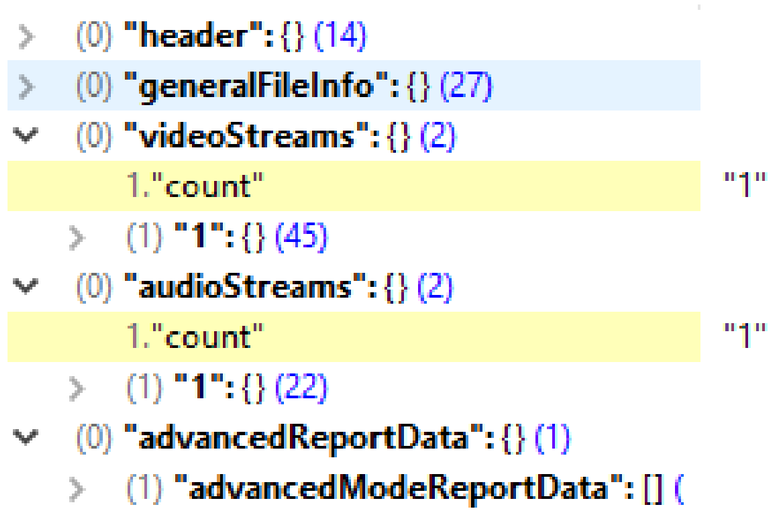
VQCSA reads VQCST test pattern file at particular workflow test point, measures VMAF scores for particular Compression Stress parameters and codecs settings, then builds Compression Stress Response Profiles

1.4 VQMA – Video Analyzer for objective video processing chain integrity tests.

VQMA module reads VQMA test pattern file at particular workflow test point, then checks multiple video parameters and checks them vs. pre-configured tolerance values

1.1 VQMINF – Media File Info Report Generator

- VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in **UTF8 JSON** format
- It can read **HDR-PQ**, **HDR-HLG** and **SDR** video as well as **LOG** video metadata
- VQMINF uses standard SHA3 and MD5 libraries to calculate unique **vqminfEssenceID** string and input file **MD5 hash** string
- Report file contains the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **textStreams**: technical parameters of text stream(s)
  - **imageStreams**: technical parameters of image stream(s)
  - **advancedReportData**: (*all raw media info strings, useful for manual analysis of complicated cases*)



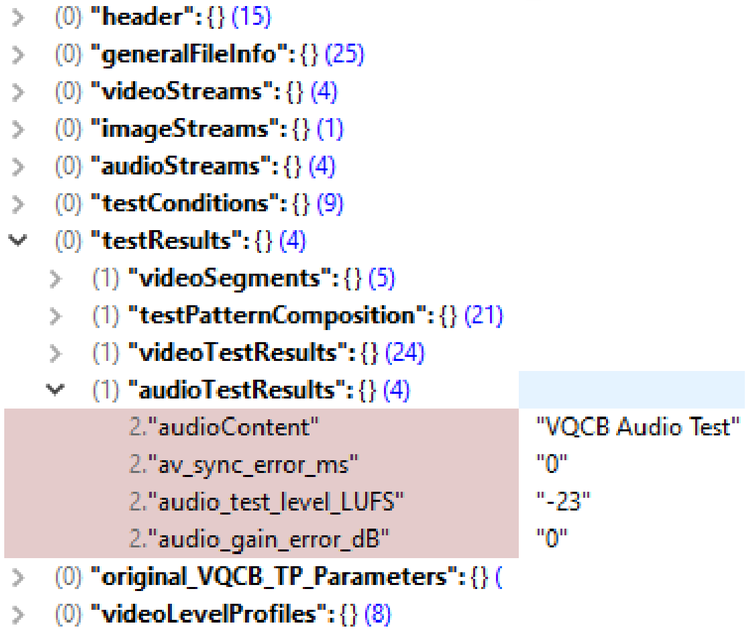
CLI interface: **vqminf** [-noard] [-nohash] -i inFilePath [-o or -o outFilePath]

1.2 VQCBA – VideoQ Color Bars Analyzer

VQCBA reads media file, containing at least one video stream (.AVI, .MOV, .MP4, etc.)

It measures video frames parameters and creates machine-readable Report in JSON format showing the following sections:

- **header**: Report timestamp and program version info
- **generalFileInfo**: container parameters, including counts of media data streams
- **videoStreams**: encoding and format information
- **imageStreams**: encoding and format information
- **audioStreams**: encoding and format information
- **testConditions**: user-selected and auto-configured test session parameters
- **testResults**: several sub-sections containing all measured parameters
- **original_VQCB_TP_Parameters**: decoded QR code string data
- **videoLevelsProfile**: pixel-by-pixel YUV and RGB video data values (*for advanced analysis*)



VQCBA can analyze VideoQ VQCB test pattern, as well as other common types of Color Bars test pattern.

CLI interface: **vqcba** [-tm | -cd] [-short] [-tfe] [-sfe] -i inFilePath [-o] or [-o outFilePath]

[-tm] flag enables 'Trailer Mode' (after main video segment), [-cd] flag enables 'Captured Data' asynchronous input mode

[-tfe] flag enables optional Thumbnail Frame Export to PNG file, [-sfe] flag enables optional Single Frame Export to lossless 16bpc MP4 file

1.2.1 VQCBA Features

- VQCBA is especially useful when streaming in **multiple formats** or when **converting** between formats
- Supported frame sizes: from **480x270** to **8K UHD**
- VQCBA auto-detects and process **5 different types** of color bars tests
- Supported color bars test patterns types (with optional audio components):
  - **VideoQ VQCB** - dynamic **HDR** and **SDR AV** test patterns
  - **ITU BT.2111 HDR-PQ** and **HDR-HLG** color bars test patterns
  - **SMPTE RP219** test pattern
  - **SMPTE EG1** legacy test pattern
  - **Full frame color bars** - the most common video test pattern
- If input file contains continuous **audio test tone**, VQCBA detects this component and measures the audio test tone **level** in dBfs; result can be found in **audioTestResults** sub-section
- In case of VideoQ VQCB test analysis, VQCBA also provides:
  - throughput **audio gain** and **AV sync error** measurements results
  - decoded VQCB QR code data, shown in the **original_VQCB_Parameters** section of the Report

1.2.2 VQCBA Report Example

- (0) **"header"**: {} (19)
- (0) **"generalInputFileInfo"**: {} (28)
- (0) **"videoStreams"**: {} (2)
- (0) **"audioStreams"**: {} (2)
- (0) **"testConditions"**: {} (10)
  - 1."timelinePositionControl" "Auto"
  - 1."selectedTimeLinePosition" "Leader"
  - 1."audioStreamAnalysis" "Yes"
  - 1."warning" "Audio and video streams durations differ"
  - 1."audioChannelsNumber" "2"
  - 1."referenceAudioChannel" "FR"
  - 1."thumbnailFileOut" "No"
  - 1."singleFrameVideoFileOut" "No"
  - 1."videoLevelProfilesReport" "Yes"
- (1) **"testCaseInitParameters"**: {} (12)
  - 2."iniFileDateTimeUTC" "2022-06-27T04:11:14.621Z"
  - 2."configuredBy" "Victor Steinberg"
  - 2."BlackLevelDelta_pct" "0.5"
  - 2."WhiteLevelDelta_pct" "0.75"
  - 2."ColorBarsLevelsDelta_pct" "0.75"
  - 2."VideoGainDelta_pct" "1"
  - 2."ColorBalanceDelta_pct" "1"
  - 2."ColorSaturationDelta_pct" "2.5"
  - 2."PLUGE_LevelsDelta_pct" "0.5"
  - 2."AudioTestToneRefLevel_dBFs" "-23"
  - 2."AudioLevelsDelta_dB" "0.75"
  - 2."AVSyncDelta_ms" "50"
- (0) **"testResults"**: {} (5)
  - (1) **"testSummary"**: {} (2)
    - 2."allTestsPassed" "Yes"
    - (2) **"partialTestsPassed"**: {} (13)
  - (1) **"videoSegments"**: {} (5)
    - 2."relevantTimelineSegments" "1"
    - 2."testPatternTimeLine" "Leader"
    - 2."analyzedFramesCount" "1200"
    - 2."analyzedDurationTC1000" "00:00:20.020"
    - (2) **"Segment1"**: {} (5)
  - (1) **"testPatternComposition"**: {} (21)
  - (1) **"videoTestResults"**: {} (26)
  - (1) **"audioTestResults"**: {} (4)
- (0) **"qrCodeBasedInfo"**: {} (2)
- (0) **"videoLevelProfiles"**: {} (8)

- (0) **"header"**: {} (19)
- (0) **"generalInputFileInfo"**: {} (28)
- (0) **"videoStreams"**: {} (2)
- (0) **"audioStreams"**: {} (2)
- (0) **"testConditions"**: {} (10)
- (0) **"testResults"**: {} (5)
  - (1) **"testSummary"**: {} (2)
    - 2."allTestsPassed" "Yes"
    - (2) **"partialTestsPassed"**: {} (13)
  - (1) **"videoSegments"**: {} (5)
  - (1) **"testPatternComposition"**: {} (21)
  - (1) **"videoTestResults"**: {} (26)
    - 2."testPatternType" "VQCB - VideoQ Color Bars"
    - 2."dynamicRangeFormat" "HDR-PQ"
    - 2."colorSpace" "YUV"
    - 2."bitsPerComponent" "10"
    - 2."dataRangeMetadata" "Narrow"
    - 2."dataRangeDetected" "Narrow"
    - 2."blackLevel" "64"
    - 2."blackLevelOffset_pct" "0"
    - 2."whiteLevelOnCB" "572"
    - 2."whiteLevelOnCB_pct" "57.99"
    - 2."blackClipOnPLUGE" "No"
    - 2."grayScaleNonLinearity_pct" "0"
    - 2."whiteClipOnGrayScale" "No"
    - 2."rangeConversionFootprint" "No"
    - 2."toneMapping" "No"
    - 2."wideColorGamutMapping" "No"
    - 2."colorMatrixMetadata" "BT.2020"
    - 2."colorMatrixDetected" "BT.2020"
    - 2."videoGainErrorOnCB_pct" "0"
    - 2."colorBalanceErrorOnCB_pct" "0"
    - 2."videoLevelsErrorOnCB_pct" "0.654999"
    - 2."saturationErrorOnCB_pct" "-2.23"
    - 2."colorMatrixingErrorFootprint" "na"
    - (2) **"colorBars"**: {} (8)
    - (2) **"plugeBars"**: {} (7)
    - (2) **"grayScale"**: {} (9)
  - (1) **"audioTestResults"**: {} (4)
    - 2."audioContent" "VQCB Audio Test"
    - 2."avsyncError_ms" "17"
    - 2."audioTestLevel_dBFs" "-23.04"
    - 2."audioGainError_dB" "-0.039999"
- (0) **"qrCodeBasedInfo"**: {} (2)
- (0) **"videoLevelProfiles"**: {} (8)

- (0) **"testResults"**: {} (5)
  - (1) **"testSummary"**: {} (2)
  - (1) **"videoSegments"**: {} (5)
  - (1) **"testPatternComposition"**: {} (21)
  - (1) **"videoTestResults"**: {} (26)
  - (1) **"audioTestResults"**: {} (4)
- (0) **"qrCodeBasedInfo"**: {} (2)
  - (1) **"originalTestPatternInfo"**: {} (16)
  - (1) **"workflowParametersInfo"**: {} (1
    - 2."analyzedParametersCount" "12"
    - 2."modifiedParametersCount" "9"
    - 2."undefinedParametersCount" "0"
    - (2) **"FrameSize"**: {} (2)
    - (2) **"TransferCharacteristics"**: {} (2
    - (2) **"ColorSpace"**: {} (2)
      - 3."original" "RGB"
      - 3."detected" "YUV"
    - (2) **"VideoDataRange"**: {} (2)
    - (2) **"SamplingStructure"**: {} (2)
      - 3."original" "444"
      - 3."detected" "420"
    - (2) **"BitsPerComponent"**: {} (2)
      - 3."original" "16"
      - 3."detected" "10"
    - (2) **"FrameRate"**: {} (2)
      - 3."original" "23.976"
      - 3."detected" "59.940"
    - (2) **"Container"**: {} (2)
      - 3."original" "MOV"
      - 3."detected" "MP4"
    - (2) **"VideoCodec"**: {} (2)
      - 3."original" "PNG"
      - 3."detected" "HEVC"
    - (2) **"AudioCodec"**: {} (2)
      - 3."original" "PCM"
      - 3."detected" "E-AC-3"
    - (2) **"AudioChannels"**: {} (2)
      - 3."original" "6"
      - 3."detected" "2"
    - (2) **"AudioSamplingRate"**: {} (2)
      - 3."original" "48000"
      - 3."detected" "44100"
- (0) **"videoLevelProfiles"**: {} (8)

1.3 VQCSA – Compression Stress Analyzer

Test conditions:

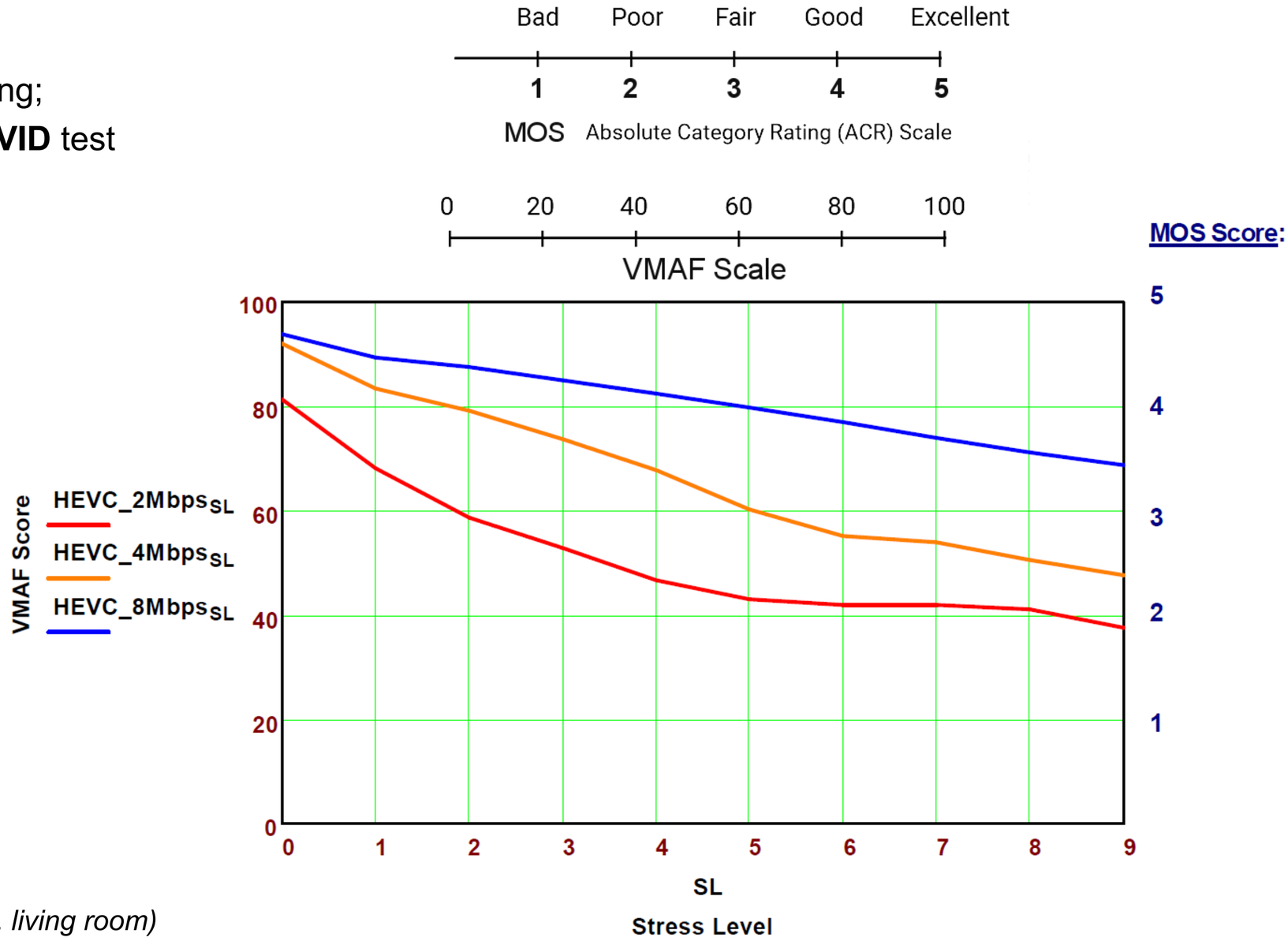
HD 60fps HEVC – **2**, **4, 8 Mbps** encoding;

Medium **S**tress **R**ange (MSR) **VQCST_VID** test

Bitrate, Mbps:

Stress Level:	2	4	8
0	81.3	92.0	93.8
1	68.1	83.4	89.3
2	58.7	79.1	87.5
3	52.8	73.7	85.0
4	46.7	67.7	82.4
5	43.0	60.2	79.7
6	42.0	55.1	77.0
7	41.9	54.0	73.9
8	41.1	50.5	71.1
9	37.6	47.6	68.7

$$VMAF \equiv \begin{pmatrix} 81.3 & 92.0 & 93.8 \\ 68.1 & 83.4 & 89.3 \\ 58.7 & 79.1 & 87.5 \\ 52.8 & 73.7 & 85.0 \\ 46.7 & 67.7 & 82.4 \\ 43.0 & 60.2 & 79.7 \\ 42.0 & 55.1 & 77.0 \\ 41.9 & 54.0 & 73.9 \\ 41.1 & 50.5 & 71.1 \\ 37.6 & 47.6 & 68.7 \end{pmatrix}$$

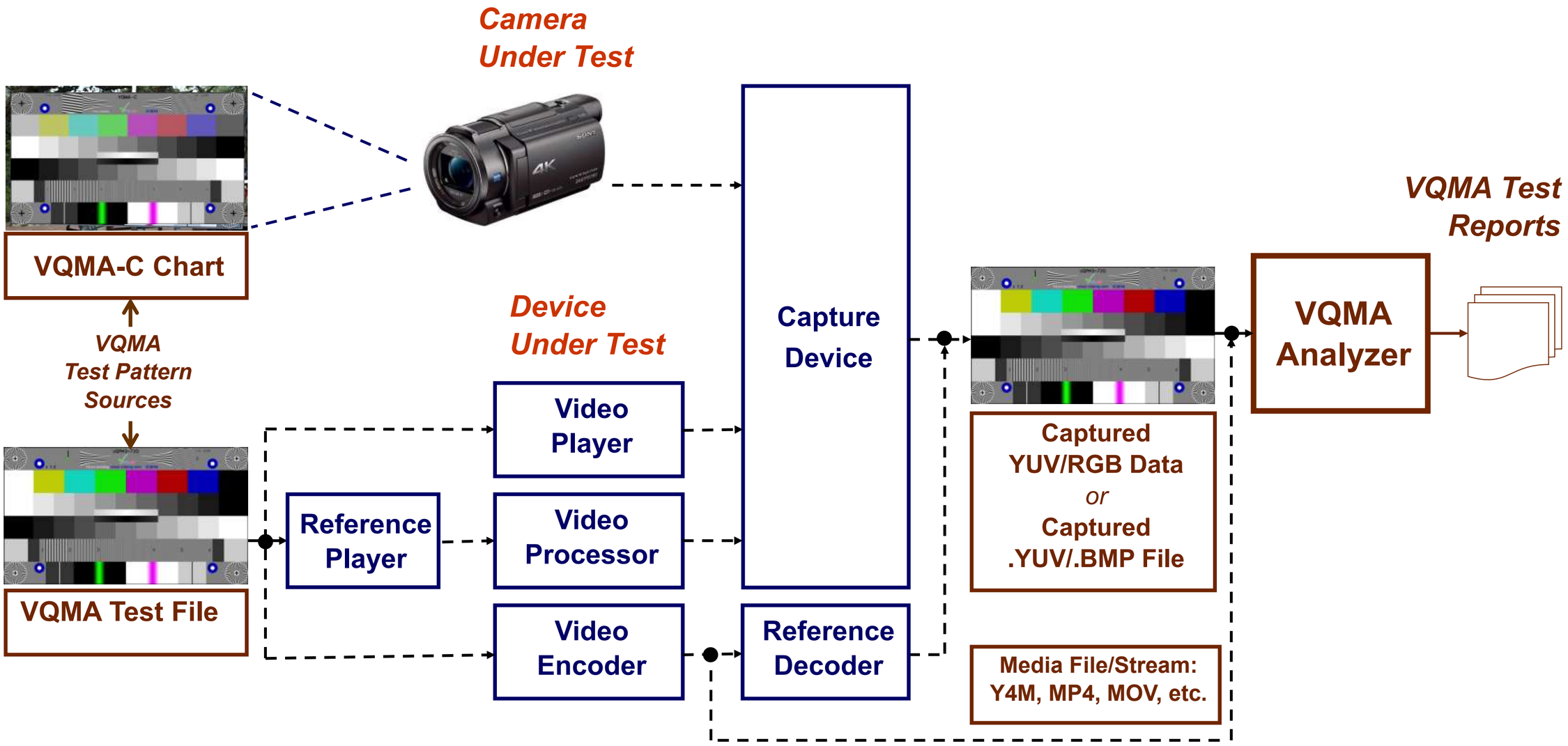


VMAF model used: Netflix vmaf_v0.6.1.pkl (HD, living room)

1.4 VQMA – Video Processing Chain Analyzer

- 4th generation of VideoQ best-selling software product,
  *suitable for any video format, any frame size from **192x108** to **4096x3072** (**4K versions**) or **7680x4320** (**8K versions**), any frame rate, **HDR & WCG support coming soon***
- Software executable under Windows™ (XP, 7, 8, 10, 11)
- USB dongle copy-protected, dongle-per-workstation
- Automated analysis on the companion VQMA Matrix Test Pattern
- Variety of VQMA Test Pattern formats: Optical Chart, File, Signal, Stream
- Unique patented algorithms for accurate & fast measurements (typically 2-5 seconds)
- Built-in YUV/RGB Waveform Scope
- Noise Measurement and Waveform Scope work on any static image
- Windows GUI Mode for R&D and product verification
- Command Line Interface (Batch) Mode for automated QA/QC operation

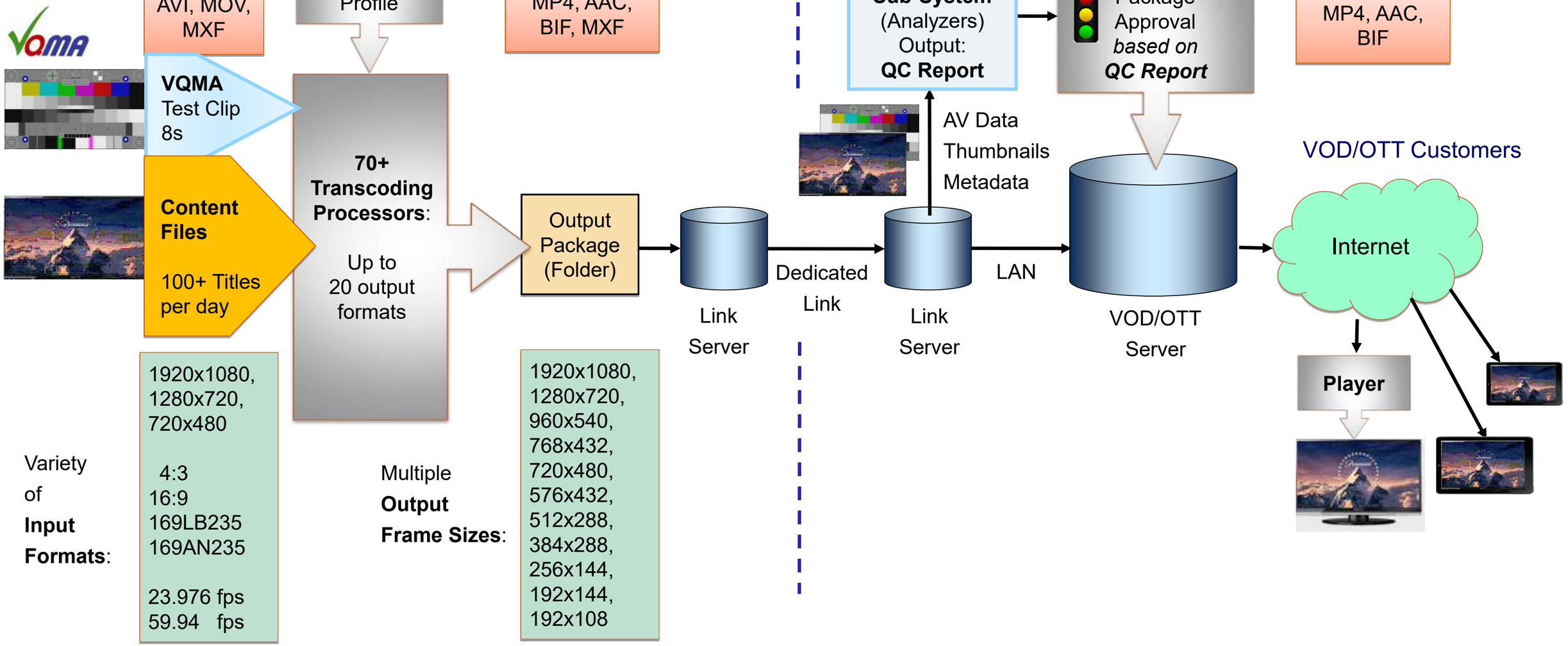
1.4.1 VQMA Workflow Variants



1.4.2 VQMA Application Example

A Large Transcoding Facility, **Los Angeles**

A Large VOD/OTT Service Facility, **Toronto**



TOC

Pack 2: AV Files Conversion, Encoding and Transcoding

P2TOC

*Click on **P2TOC** in the upper-right corner for **Pack 2 Table Of Contents***

2.1 VQMINF – Media File Info Report Generator

VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in UTF8 JSON format.

2.2 VQBIF – BIF (Base Index Frames) Files Verifier

VQBIF reads BIF thumbnail files, checks the parameters for compliance, and extracts numbered JPEG files as a reference decoder

2.3 VQBLA – Bitrate Ladder Analyzer

VQBLA reads input media file, then checks multiple video parameters and builds the encoding quality score profiles for a given codec and preconfigured set of output frame sizes and bitrates

2.4 VQC – HDR-SDR Converter

2.5 VQCSA – Compression Stress Analyzer, companion program for VQCST test patterns.

VQCSA reads VQCST test pattern file at particular workflow test point, measures VMAF scores for particular Compression Stress parameters and codecs settings, then builds Compression Stress Response Profiles

2.6 VQLPN – Audio Loudness Profiler and Normalizer

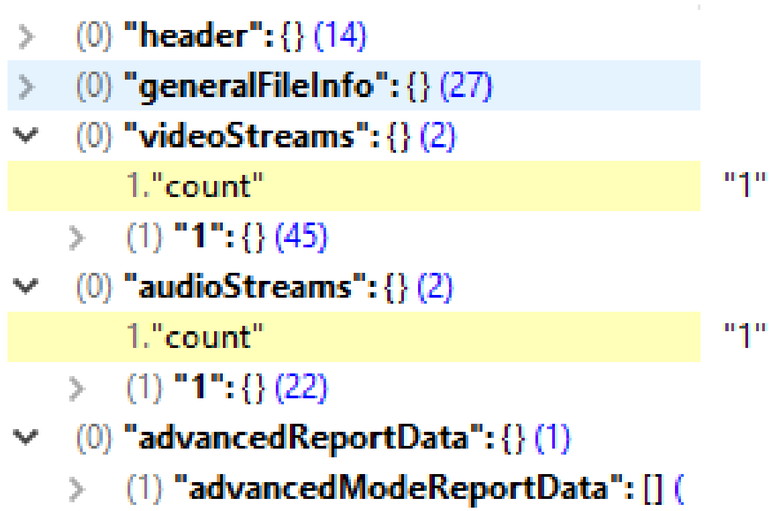
VQLPN measures and normalizes the audio stream loudness parameters in accordance with ITU-R BS.1770-4 (USA ATSC RP A85, EBU R128).

2.7 VQTSF – Transcoding Segments Finder

VQTSF reads input media file, detects critical timeline segments positions, measures content activities and builds related timeline profiles

2.1 VQMINF – Media File Info Report Generator

- VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in **UTF8 JSON** format
- It can read **HDR-PQ**, **HDR-HLG** and **SDR** video as well as **LOG** video metadata
- VQMINF uses standard SHA3 and MD5 libraries to calculate unique **vqminfEssenceID** string and input file **MD5 hash** string
- Report file contains the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **textStreams**: technical parameters of text stream(s)
  - **imageStreams**: technical parameters of image stream(s)
  - **advancedReportData**: (*all raw media info strings, useful for manual analysis of complicated cases*)



CLI interface: **vqminf** [-noard] [-nohash] -i inFilePath [-o or -o outFilePath]

2.2. VQBIF – BIF (Base Index Frames) Files Verifier

- VQBIF reads 1, 2 or 3 .BIF thumbnails files, and creates Report in JSON format.
- Validate BIF files for:
  - File integrity
  - Size [each frame/picture]
    - For high bitrate it should be …
    - For low bitrate it should be …
  - Coding quality [jpeg compression quality]
    - For high bitrate it should be …
    - For low bitrate it should be …
  - Offset (time offset between each grab)

- Check for **time offset correlation** (*function of the specified media file duration*)
- Check for **cumulative amount** of index frame images
- **Decode** the BIF file(s) as a **reference decoder** (*extract several sets of numbered .JPG files*)

CLI interface: **vqbif** -i in1.BIF [in2.BIF] [in3.BIF] [-o outPath] [-j/-q] [jpeg_folder] [-r Ref_Framewise_Separation_ms]

2.3 VQBLA – Bitrate Ladder Analyzer

- VQBLA reads .MOV, MP4, etc. media file, and analyzes its content enabling optimal downstream transcoder settings
- It measures Intra-frame and Inter-frame Activities
- VQBLA builds statistics (histograms) for a range of critical video parameters
- The key stage of VQBLA work is estimation of **expected levels of compression artifacts** and **down-scaling distortions**
- Finally, VQBLA creates machine-readable Report in JSON format showing:
  - Measured video parameters statistics: **Intra-frame Activity** and **Inter-frame Activity**
  - The BLA **Draft Bitrate Ladder** as an array of the expected **Quality Score** vs. **Bitrate** and **Frame Size**
- VQBLA also creates an optional .PNG image file showing (see next slide):
  - Activities Timeline Profile Plot
  - Activities Statistics Bargraphs
  - Bitrate Ladder Plot with critical bitrate values marked

CLI interface: **vqbla** [-cs NumOfSec] [-p] -i inFilePath [-o [outFilePath]

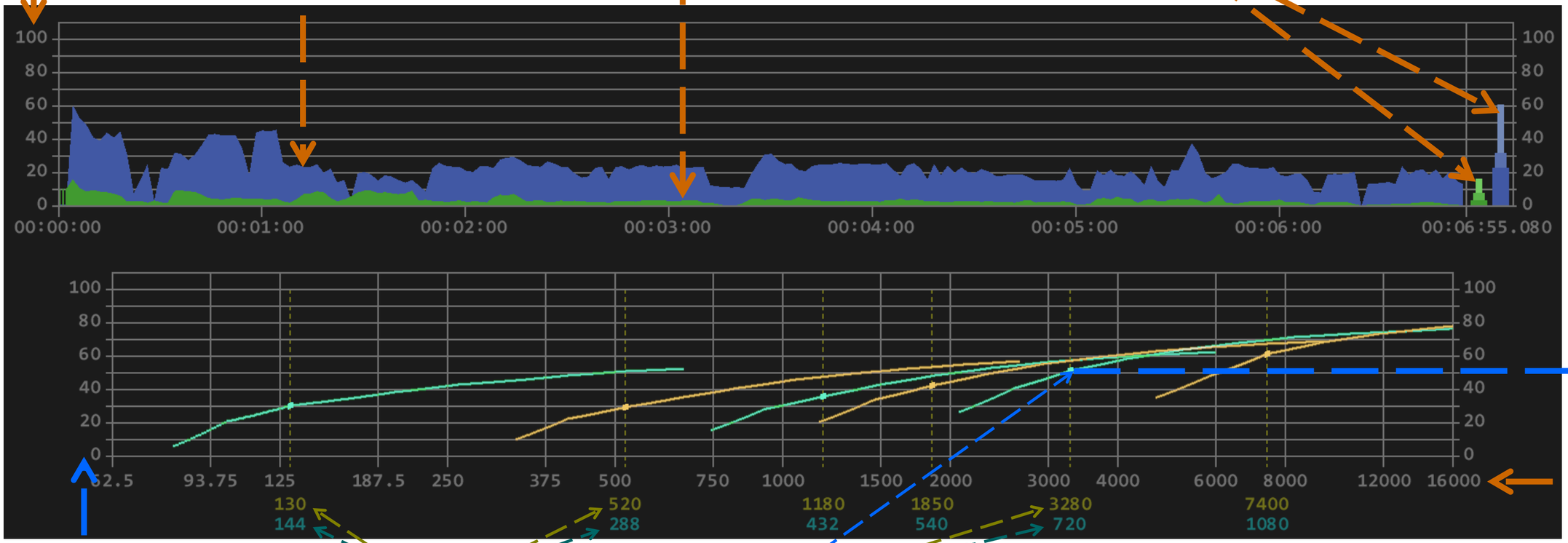
2.3.1 VQBLA Plot Example

Activity, %

Intra-frame Activity
(Static Details)

Inter-frame Activity
(Dynamic Details)

Activities Statistics Bargraphs



Quality Score
0 .. 100

Bitrate, kbps

Quality Score

Bitrate Ladder Items:
Critical Bitrates, kbps
vs. Frame Heights

Quality Score
and Critical Bitrate, kbps
for this Frame Height

Encoding with bitrates lower than critical results in compression artifacts, usually described as "Noticeable and annoying", bitrates above critical – as "Noticeable, but not annoying"

2.4 VQC – HDR-SDR Converter

- VQC reads media file or image files sequence, then converts input file HDR-PQ, HDR-HLG, or SDR **dynamic range** and **color space format** to the specified output format
- Frame sizes: from **1280x720** to **8K UHD**
- Supported input and output color primaries:
  - **BT.709** (aka NCG = Narrow Color Gamut, *only for SDR*)
  - **BT.2020** (aka WCG = Wide Color Gamut)
  - **P3** (SMPTE432, aka ECG = Expanded Color Gamut)
- VQC internal workflow consists of several stages:
  - **Reading** media file or image files sequence
  - **Measurement** of video frames **parameters**
  - **Conversion** of video data to the specified **dynamic range** and **color space** format
  - **Encoding** of video data to the specified output file(s) format
  - Creation of machine-readable **JSON Report** and optional **PNG Plot** file

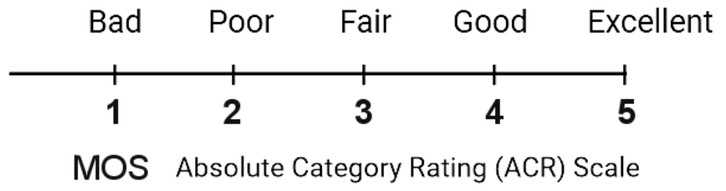
CLI interface: **vqc** [-c configFilePath] -i inFilePath -o outFilePath

2.5 VQCSA – Compression Stress Analyzer

Test conditions:

HD 60fps HEVC – **2**, **4, 8 Mbps** encoding;

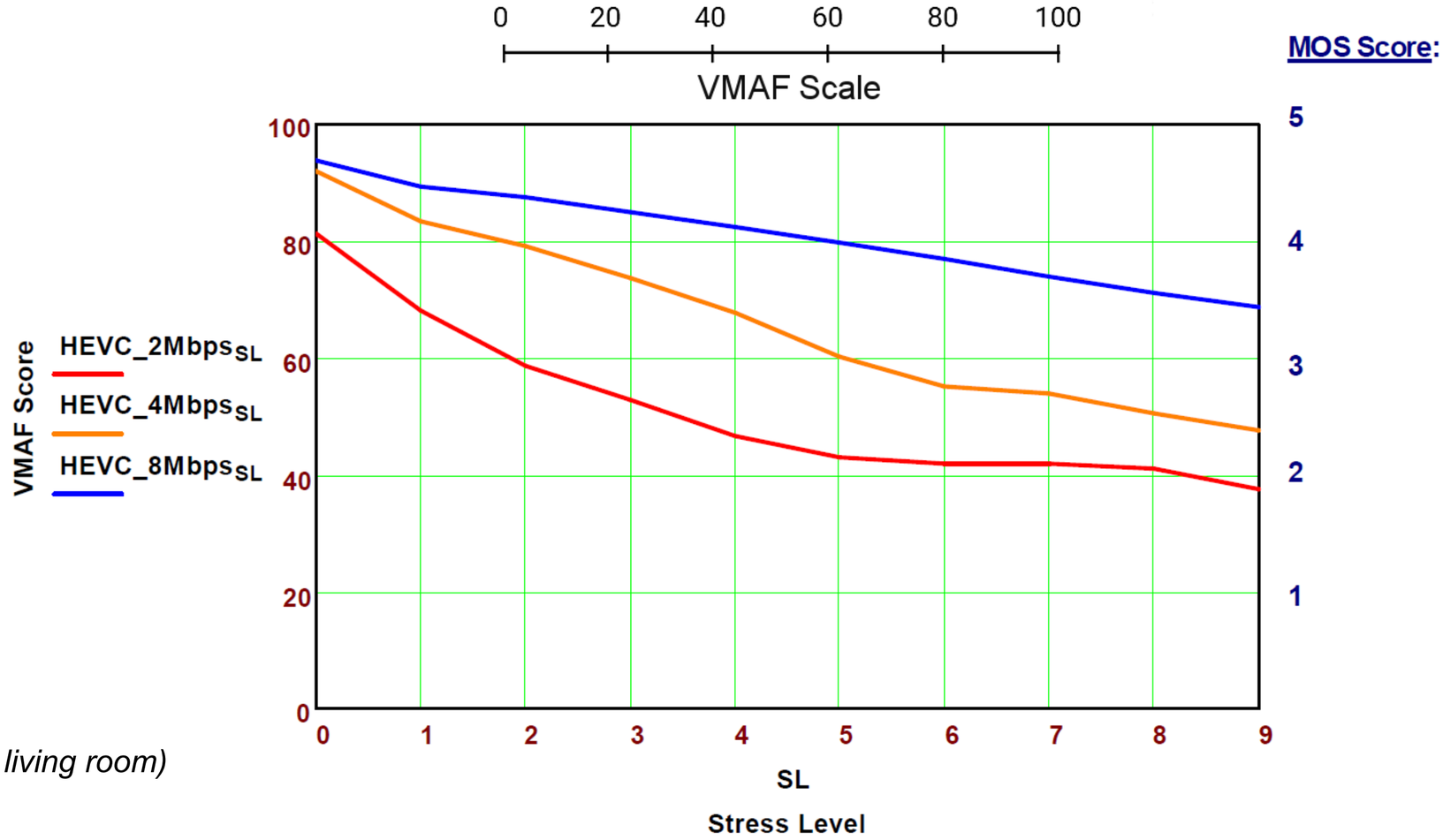
Medium **S**tress **R**ange (MSR) **VQCST_VID** test



Bitrate, Mbps:

	2	4	8	Stress Level:
VMAF ≡	81.3	92.0	93.8	0
	68.1	83.4	89.3	1
	58.7	79.1	87.5	2
	52.8	73.7	85.0	3
	46.7	67.7	82.4	4
	43.0	60.2	79.7	5
	42.0	55.1	77.0	6
	41.9	54.0	73.9	7
	41.1	50.5	71.1	8
	37.6	47.6	68.7	9

VMAF model used: Netflix vmaf_v0.6.1.pkl (HD, living room)



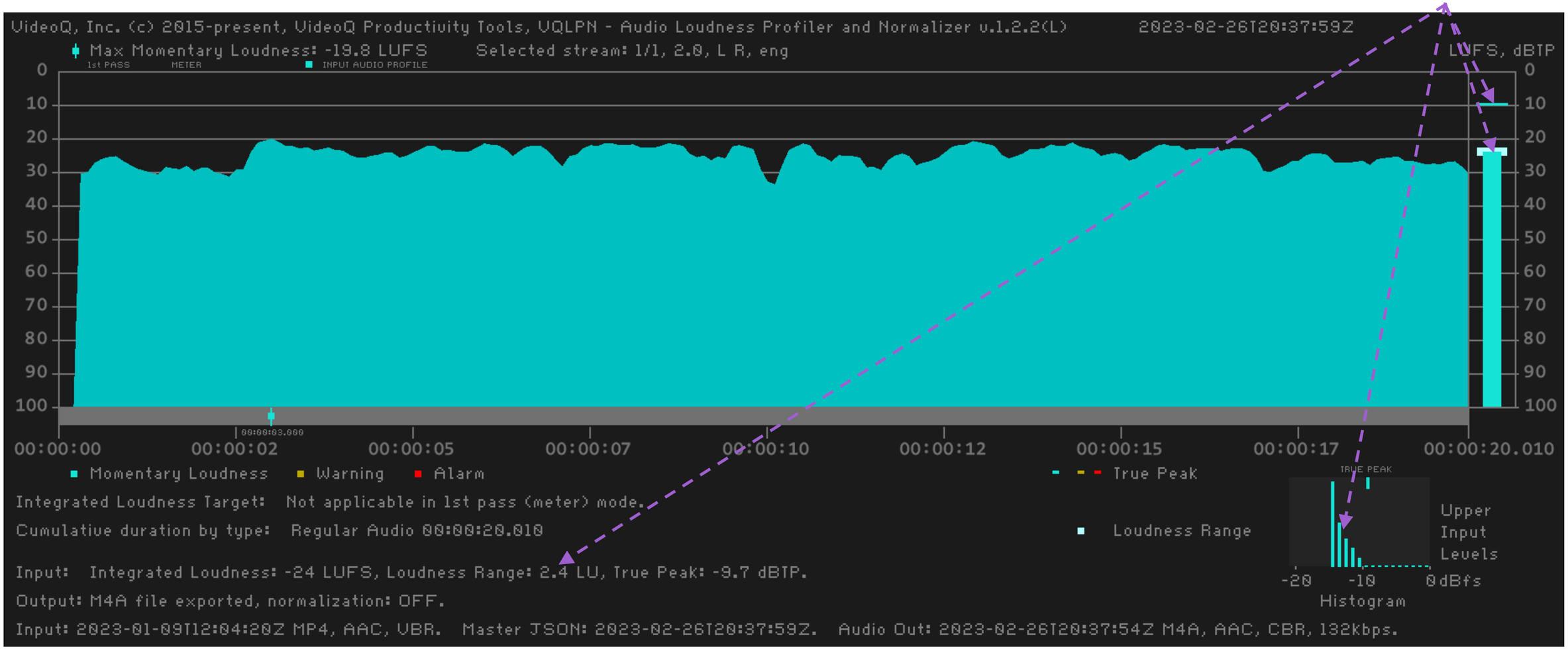
2.6 VQLPN – Audio Loudness Profiler & Normalizer

- VQLPN reads media file, containing audio stream(s), measures and optionally normalizes audio loudness.
- VQLPN supports MP4, MOV, MXF, WAV, W64, AAC, AC3, EAC3, etc., and various audio stream formats:
  - any bit depth, bit rate and sampling rate, all audio codecs supported by ffmpeg
  - multi-channel and multi-track formats: **1.0, 2.0, 5.1, 7.1**
- It measures the audio stream loudness parameters in accordance with Recommendation ITU-R **BS.1770-4** (*USA **ATSC RP A85**, EBU **R128***). Editable *.INI file stores test configuration and target parameters.
- Sorts audio segments by types (**regular audio, mute, silence, test tone, test pattern**)
- Finally, VQLPN creates detailed Report in JSON format, including **Momentary Loudness Profile** data array at 100 ms step interval
- Configurable outputs:
  - **Audio file** in the **desired format,** optionally **normalized** to the desired **Integrated Loudness** target
  - **PNG image file** showing **momentary loudness** time-line profile **plot**, **loudness** statistics **bargraph**, **upper levels histogram**, as well as other useful markers and values
- Optional stand-alone utility modules: **VQLPC** *Correlator*, **VQLPP** *Plotter*, **VQILM** *IL Meter*

CLI interface: **vqlpn** [-j jsonFilePath] [-c configFilePath] [-br bitRate_kbps] -i inFilePath [-o [outFilePath]]

2.6.1 VQLPN Plot Example 1

- Professional advert clip with very short mute segment at start, measured (checked) by VQLPN in “Meter” mode
- **Normalized audio stream** – **Integrated Loudness** is exactly equal to **-24 LUFS** ATSC standard value
- **Loudness Range 2.4 LU** is rather **low**, but for this sort of content it is **normal**
- **True Peak** value is **good**, and **Upper Levels Histogram** looks **good**

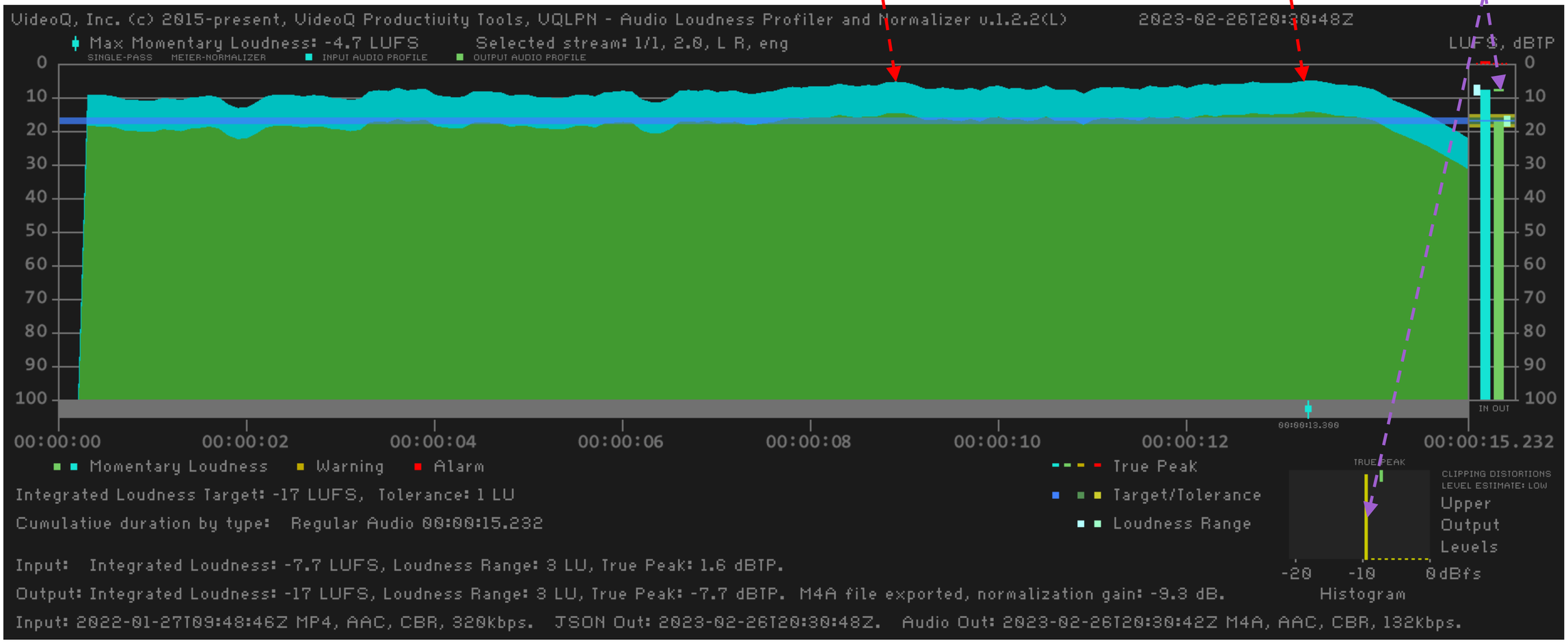


2.6.2 VQLPN Plot Example 2

- **Extremely loud** audio source **Integrated Loudness** after normalization is **exactly equal** to **-17 LUFS** (webcasting) target
- **Output True Peak** value looks **good**, but **Upper Levels Histogram** still shows **Low** level of possible **Clipping Distortions**

Conclusion: *The normalized clip is suitable for webcasting, probably with minor clipping distortions issues*

***True Peak** level now **looks good**, but **Upper Levels Histogram** still shows possible problems*



2.6.3 VQLPN Plot Example 3

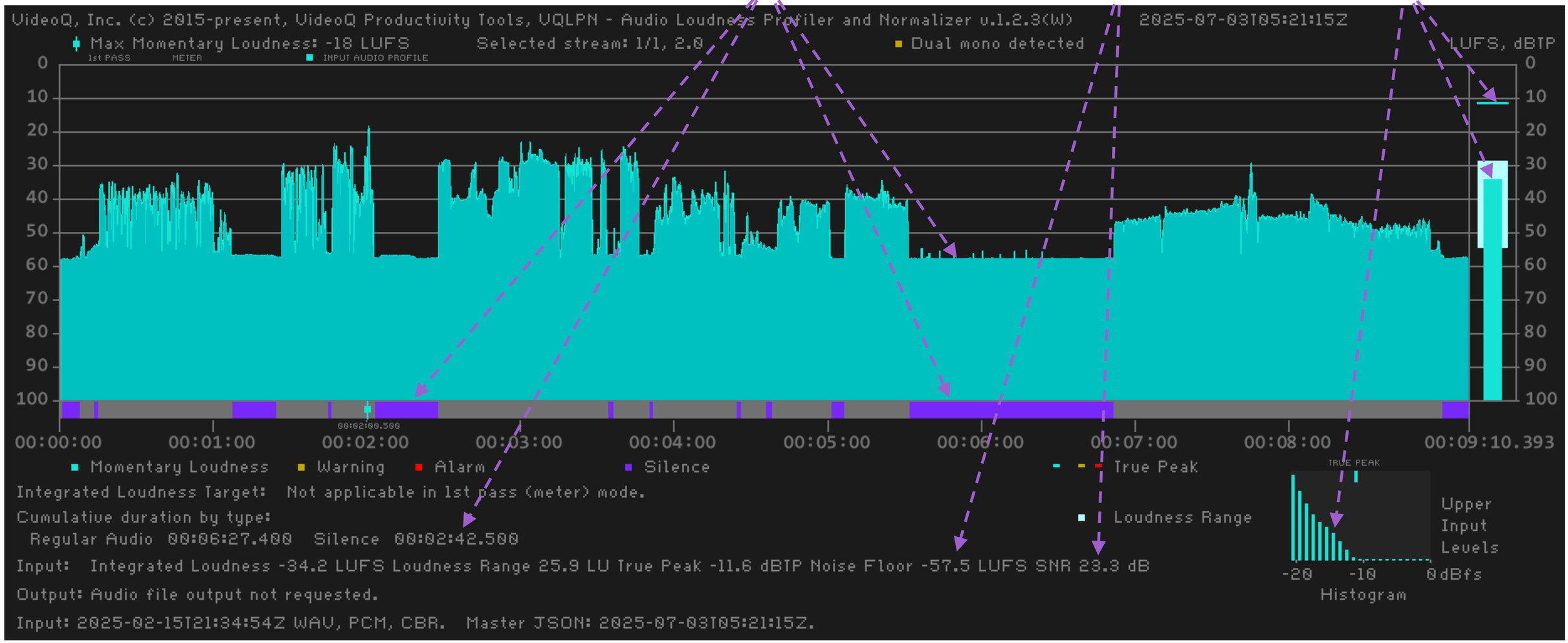
- **Scanned** 16mm film MAG track **Integrated Loudness** is **extremely low: -34.2 LUFS** (*far below all target values*)
- **True Peak** value looks rather **high** for measured low IL; this may lead to **Clipping Distortions** *after high normalization gain*

Conclusion:
*Strong **level correction** and AV timeline **editing** required*

***Silence** segments take about 30% of duration*

Noise Floor** and **SNR** values are rather **good

***IL Bar** and **Histogram** show **extremely low** levels*



2.7 VQTSF – Transcoding Segments Finder

- VQTSF reads .MOV, MP4, etc. media file, and analyzes its content enabling optimal downstream transcoder settings. This is achieved via detection of critical timeline segments positions (**preferred key frames**). VQTSF also measures **content activities** and builds related timeline profile.
- VQTSF creates machine-readable Report in JSON format showing the following sections:
  - **header**
  - **generalFileInfo**
  - **videoStream**:
  - **videoSegments**
  - **activitiesStatistics**_dBfs
    - spatial
    - temporal
  - **timeLineActivitiesProfiles**_dBfs
    - spatial
    - temporal

CLI interface: **vqtsf** [-noplot] [-slo] -i inFilePath [-o [outFilePath]]

2.7.1 VQTSF Plot Example

File duration: 6min 55s. 74 segments found, segment durations from 0.64s to 25.2s.

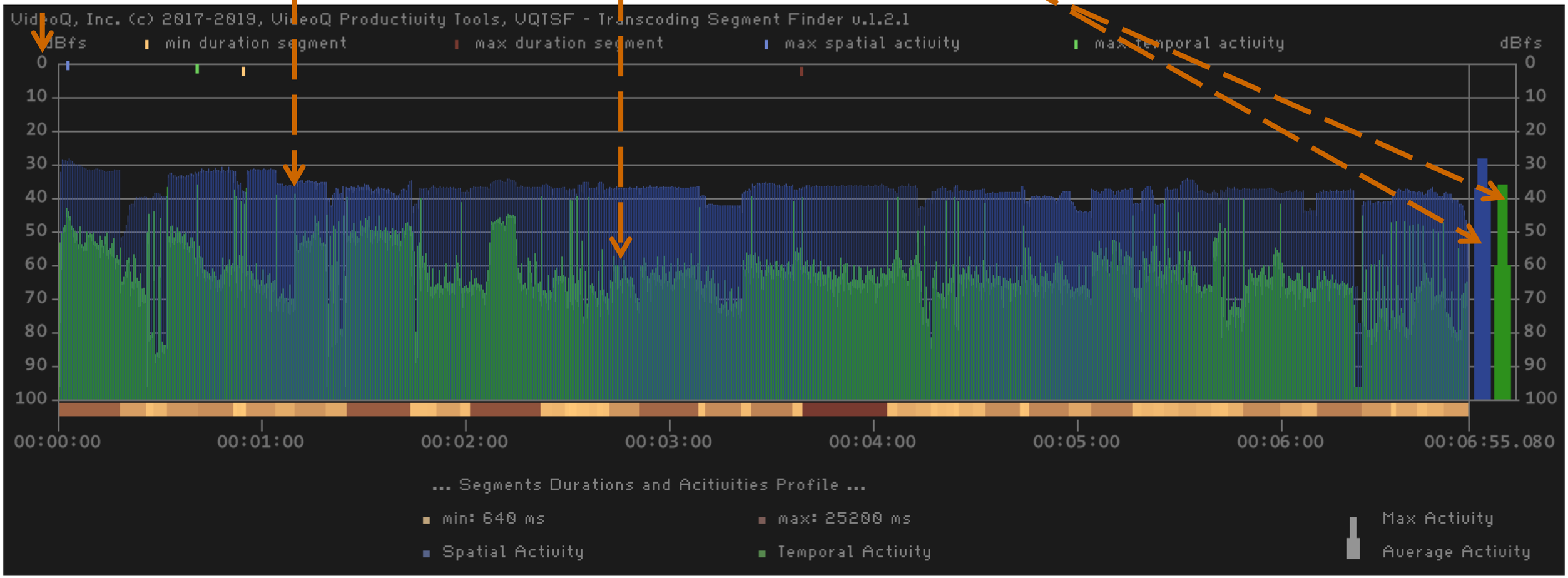
Activity *profiles are of* ***medium*** *strength, so we can get relatively* ***good quality*** *at relatively* ***low bitrates****.*

Activity, %

Intra-frame Activity
(Static Details)

Inter-frame Activity
(Dynamic Details)

Activities Statistics Bargraphs



Pack 3: VQPT Modules for AV Content Tests

P3TOC

*Click on **P3TOC** in the upper-right corner for **Pack 3 Table Of Contents***

3.1 VQMINF – Media File Info Report Generator

VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in UTF8 JSON format.
It can read HDR-PQ, HDR-HLG and SDR video as well as LOG video metadata.
VQMINF also calculates unique vqminfEssenceID string and input file MD5 hash string.

3.2 VQCFA – Captions Files Analyzer

*VQCFA reads .**WebVTT**, .**VTT**, .**XML** or .**TTML** caption file, measures **caption parameters** and checks them against the predefined, auto-switchable, and/or CLI specified, thresholds; VQCFA also exports full captions event listing text file.*

3.3 VQFP – Video Frames Profiler

*VQFP reads media file, measures video frames active image formats, frame cadences, sharpness, noise level, video data and light level statistics, detects video segments, and calculate **timeline profiles** of video levels, light level, sharpness and details activity.*

3.4 VQLPC – Loudness Profiles Correlator, companion program for VQLPN module

*VQLPC reads two JSON report files created by **VQLPN** module (Reference and Test inputs), detects and compares momentary loudness time-line profiles of two inputs. Finally, VQLPC calculates the **correlation of two inputs**.*

3.5 VQLPN – Audio Loudness Profiler and Normalizer

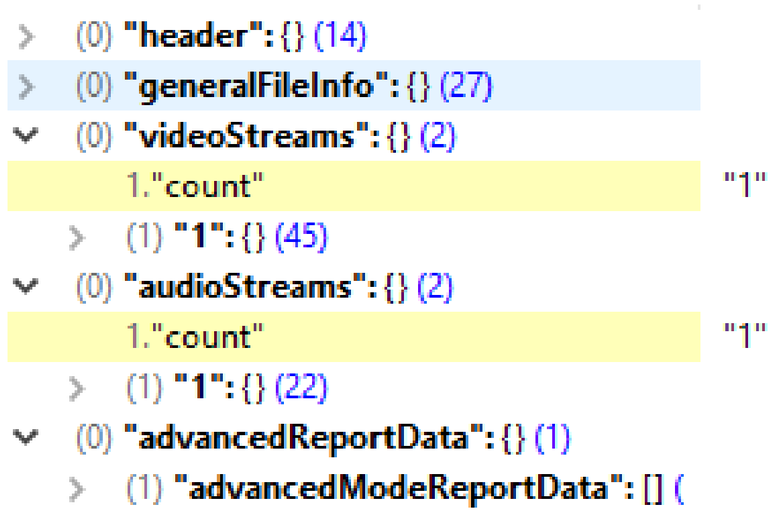
It measures and normalizes the audio stream loudness parameters in accordance with ITU-R BS.1770-4 (USA ATSC RP A85, EBU R128).

3.6 VQPLA – Picture Levels Analyzer

*VQPLA reads SDR/HDR media file, detects video segments, and calculates **timeline profiles** of **video and light levels**.*

3.1 VQMINF – Media File Info Report Generator

- VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in **UTF8 JSON** format
- It can read **HDR-PQ**, **HDR-HLG** and **SDR** video as well as **LOG** video metadata
- VQMINF uses standard SHA3 and MD5 libraries to calculate unique **vqminfEssenceID** string and input file **MD5 hash** string
- Report file contains the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **textStreams**: technical parameters of text stream(s)
  - **imageStreams**: technical parameters of image stream(s)
  - **advancedReportData**: (*all raw media info strings, useful for manual analysis of complicated cases*)



CLI interface: **vqminf** [-noard] [-nohash] -i inFilePath [-o or -o outFilePath]

3.2 VQCFA – Captions Files Analyzer

- VQCFA reads .**WebVTT**, .**VTT**, .**XML** or .**TTML** caption file
- It measures **caption parameters** and checks them against the predefined, auto-switchable, and/or CLI specified, **thresholds**.
- Finally, VQCFA creates machine-readable Report in JSON format showing the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **testConditions**: user-selected and auto-configured test session parameters
  - **eventsStatistics**: measured statistics
  - **invalidEvents**, which is further sub-divided into two sub-sections:
    - invalidEventsByType
    - invalidEventsByNumber
  - **Full Captions Event Listing** (*decoded unformatted UTF text lines with timestamps*)

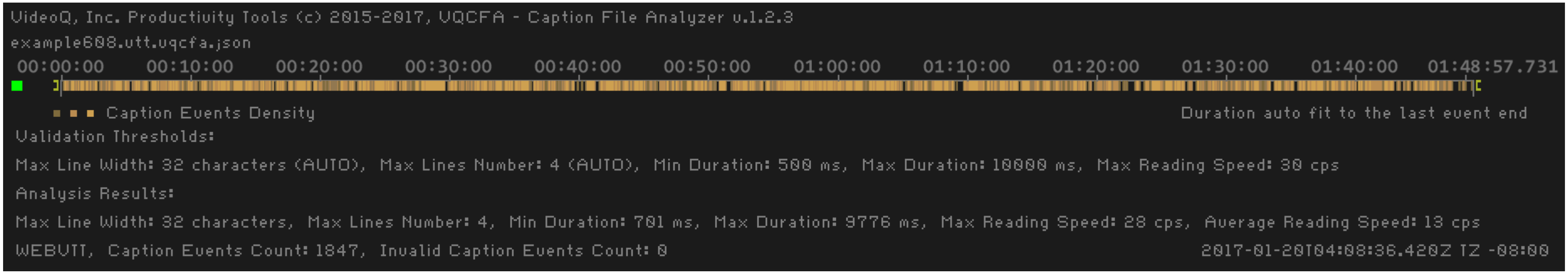
CLI interface: **vqcfa** [-p int int] [-lw int -ln int] -i inFilePath [-o [outFilePath]]

3.2.1 VQCFA Report Example

- (0) **"header"**: {} (11)
- (0) **"generalInfo"**: {} (21)
- (0) **"testConditions"**: {} (14)
- (0) **"eventsStatistics"**: {} (18)
- (0) **"invalidEvents"**: {} (3)
  - 1."invalidEventsCount" "9"
  - (1) **"invalidEventsByType"**: {} (9)
    - 2."lineWidthAboveLimit" "1"
    - 2."linesNumberAboveLimit" "1"
    - 2."durationBelowLimit" "1"
    - 2."durationAboveLimit" "1"
    - 2."readingSpeedAboveLimit" "3"
    - 2."reversedStartEndTime" "1"
    - 2."overlappingTimePositions" "1"
    - 2."beyondVideoDuration" "0"
    - 2."lateGlobalStartTime" "0"
  - (1) **"invalidEventsByNumber"**: [] (9)
- (0) **"captionEventsListing"**: {} (3)
  - 1."eventsCount" "1393"
  - 1."duration" "02:09:35.617"
  - (1) **"byTimePosition"**: [] (1393)
    - 2. 0 "00:00:00.500 --> 00:00:03.000 | VideoQ CC Test No 1 | (c) Copyright 2016. VideoQ, Inc."
    - 2. 1 "00:00:03.000 --> 00:00:04.750 | Part 1 | Suite of Basic Tests"
    - 2. 2 "00:00:04.750 --> 00:00:07.500 | suitable for original VTT files | (not for VTT converted from SCC)"
    - 2. 3 "00:00:07.500 --> 00:00:09.500 | Max Characters per Line: 42 | Max Lines Number: 2"
    - 2. 4 "00:00:10.000 --> 00:00:11.000 | Line Width Check Tests:"
    - 2. 5 "00:00:11.000 --> 00:00:13.000 | #1 Valid: 42 characters | #2 Invalid: 43 characters"
    - 2. 6 "00:00:13.000 --> 00:00:16.000 | Valid Line Width: 42 characters | 0123456789 01 Last Character Position = 42"
    - 2. 7 "00:00:16.000 --> 00:00:18.500 | Invalid Line Width: 43 characters | 0123456789 012 Last Character Position = 43"
    - 2. 8 "00:00:19.500 --> 00:00:21.000 | Lines Number Check Tests:"
    - 2. 9 "00:00:21.000 --> 00:00:23.500 | #1 Valid Number: 2 Lines | #2 Invalid Number: 3 Lines"
    - 2. 10 "00:00:23.500 --> 00:00:26.000 | Valid: Total = 2 Lines | Line 2: Valid"

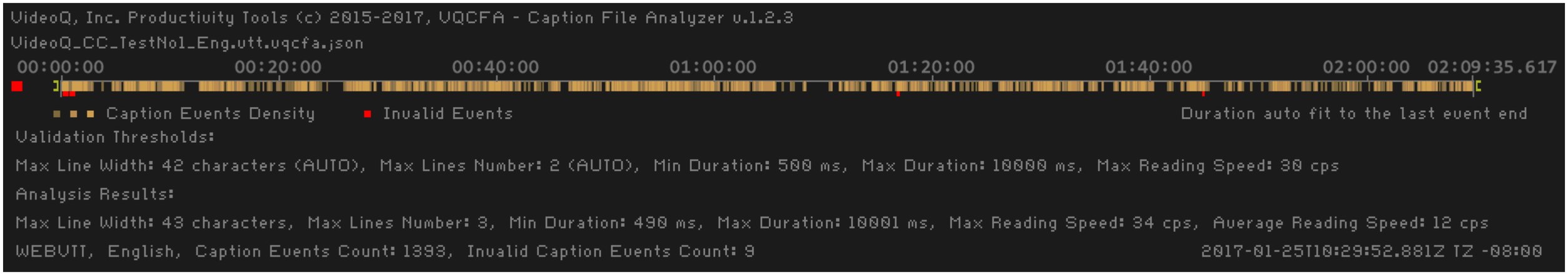
3.2.2 VQCFA Plot Examples

Normal Caption Events – **No problems found**



Multiple Caption Events are **Out of Specs:**

Reading Speed, Min Duration, Max Duration,
Overlapping Events, Max Lines Number, Max Chars Per Line

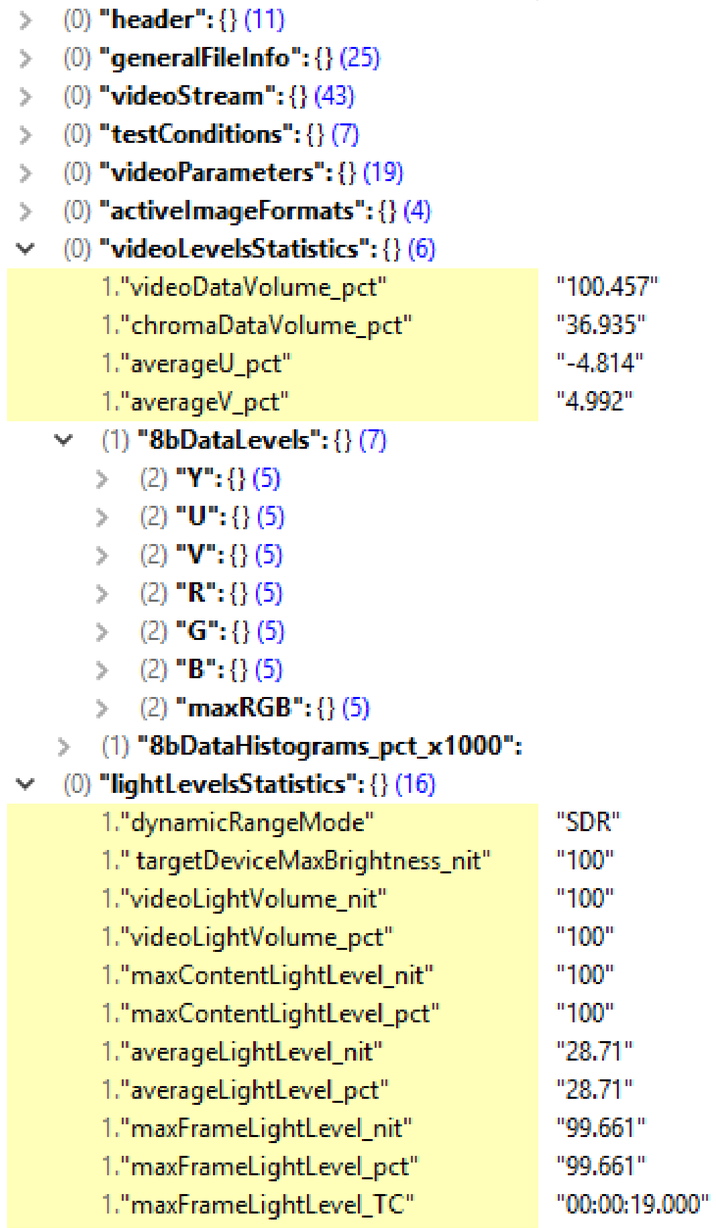


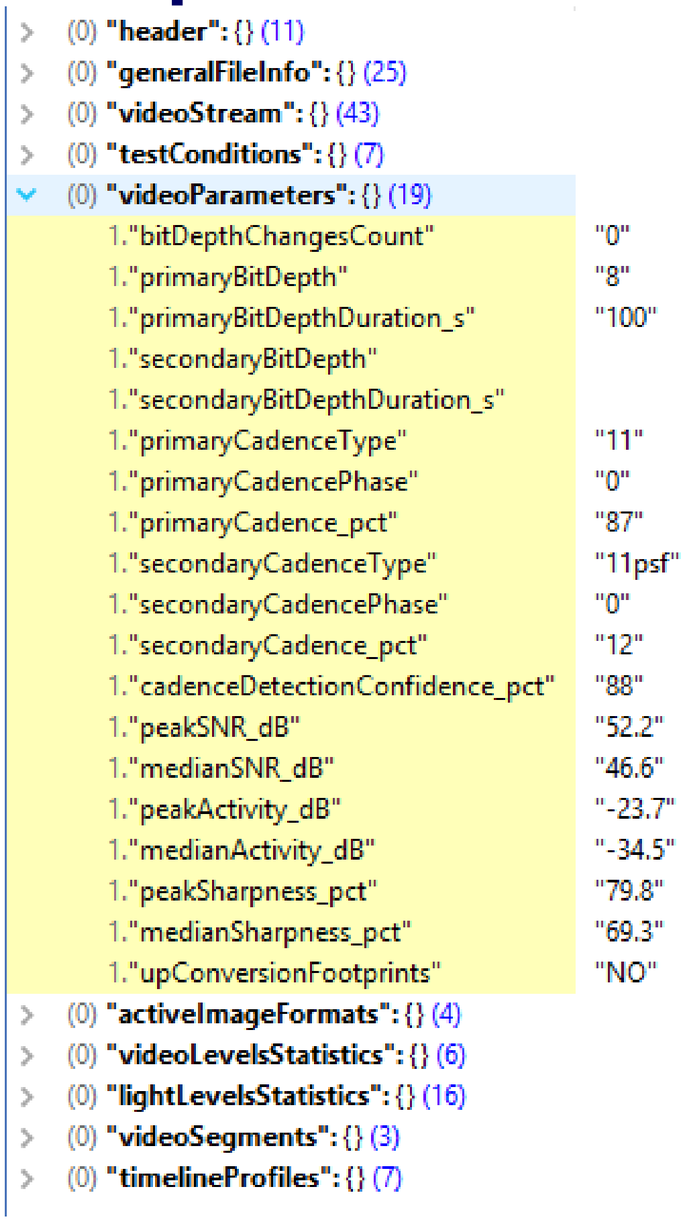
3.3 VQFP – Video Frames Profiler

- VQFP reads media file, containing at least one video stream (.MOV, .MP4, etc.)
- It measures video frames parameters and creates machine-readable Report in JSON format showing the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStream**: encoding and format information
  - **testConditions**: user-selected and auto-configured test session parameters
  - **videoParameters**: bit depth variations, frame cadencies, SNR, sharpness, details activity, up-conversion footprints
  - **activeImageFormats**: integrated durations and active image sizes of all detected active image formats (**LetterBox**, **PillarBox**, etc.)
  - **videoLevelsStatistics**: Average and Max values in % and global histograms for Y,U,V,R,G,B and MaxRGB channels
  - **lightLevelsStatistics**: Average and Max values in nits and % of the specified TDMB
  - **videoSegments**, sorted by type and by number (in order of appearance)
  - **timeLineProfiles** of video levels, light level, sharpness and details activity

CLI interface: **vqfp** [-noplot] [-short] [-vsi] [-nr | -fr] [-sdr | -pq | -hlg] -i inFilePath [-o] or [-o outFilePath]

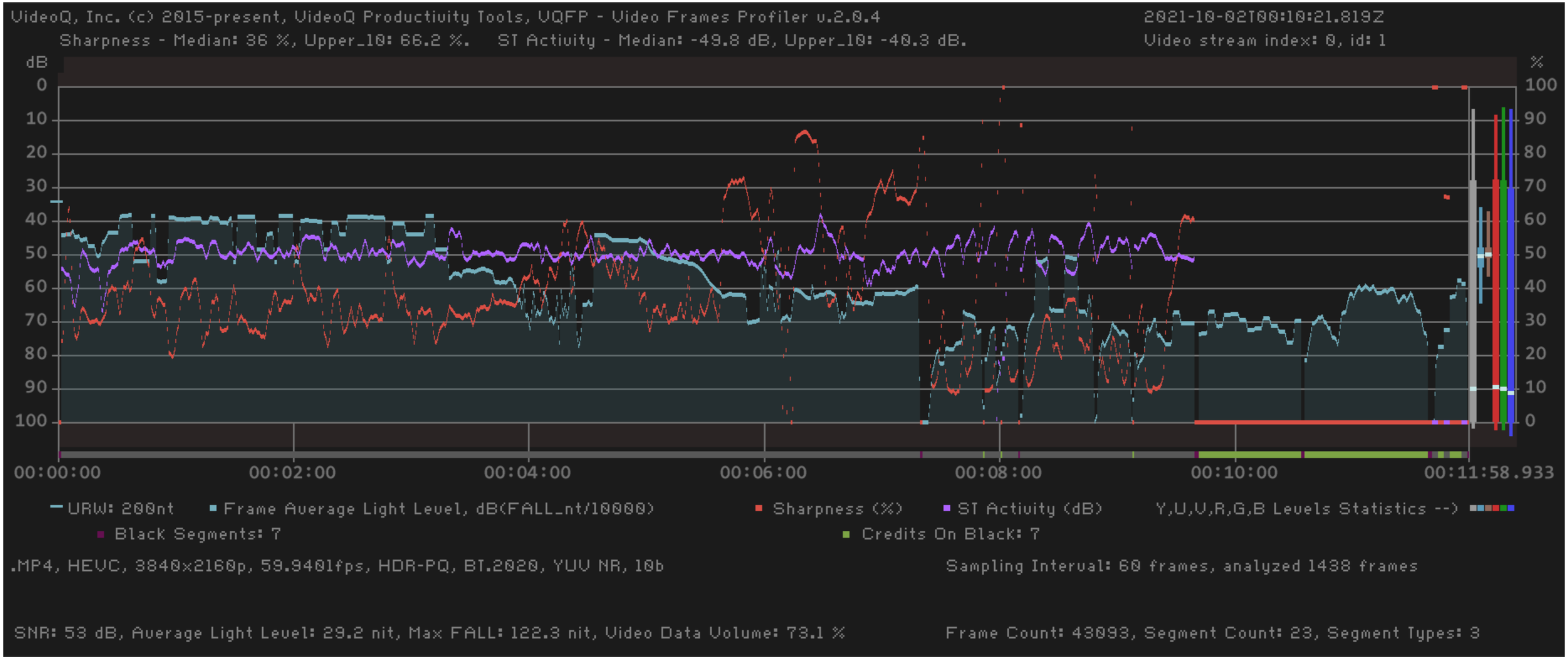
3.3.1 VQFP JSON Report Examples





3.3.2 VQFP Plot Example

Typical HDR10 clip with 2 minutes long Credits On Black at the end
*Relatively **low** spatio-temporal **activity** and **medium** **sharpness***



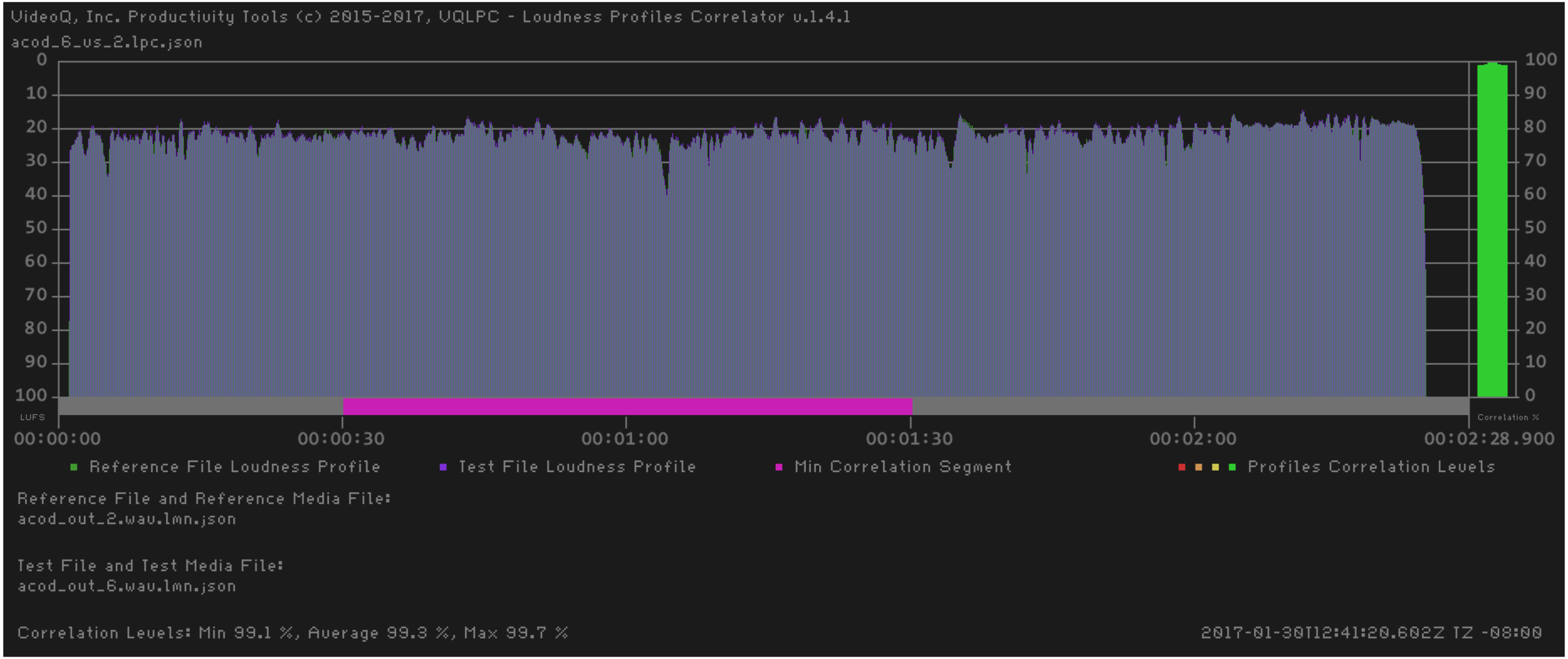
3.4 VQLPC – Loudness Profiles Correlator

- VQLPC reads **two** JSON report files created by **VQLPN** module (**Reference** and **Test** inputs)
- It detects and compares **momentary loudness** time-line profiles of two inputs, then creates Report in JSON format
- The most important output parameter is the **correlation value** in %:
  - Value of 100.0 means **perfect match**
  - Values above 99.2 means **small discrepancy**
  - Values below 85.0 mean **significant discrepancy**,
    e.g. caused by two different dialog languages combined with the same noise and music international soundtrack
  - Values below 55 (and down to 0) mean **very strong differences**, *e.g. two different episodes or two different clips*
- Optional output:
  - **PNG file** showing both inputs loudness time-line profiles plots, as well as global correlation parameters BarGraph

CLI interface: **vqlpc** [-p] -i refFilePath testFilePath [-o or -o outFilePath]

3.4.1 VQLPC Plot Example 1

Two inputs are two different versions (2.0 and 5.1) of the **same audio track**:
correlation is very high – about 100%



3.4.2 VQLPC Plot Example 2

Two inputs are in fact **two different audio tracks**
Loudness profiles and durations look similar, but actual **correlation value is very low**

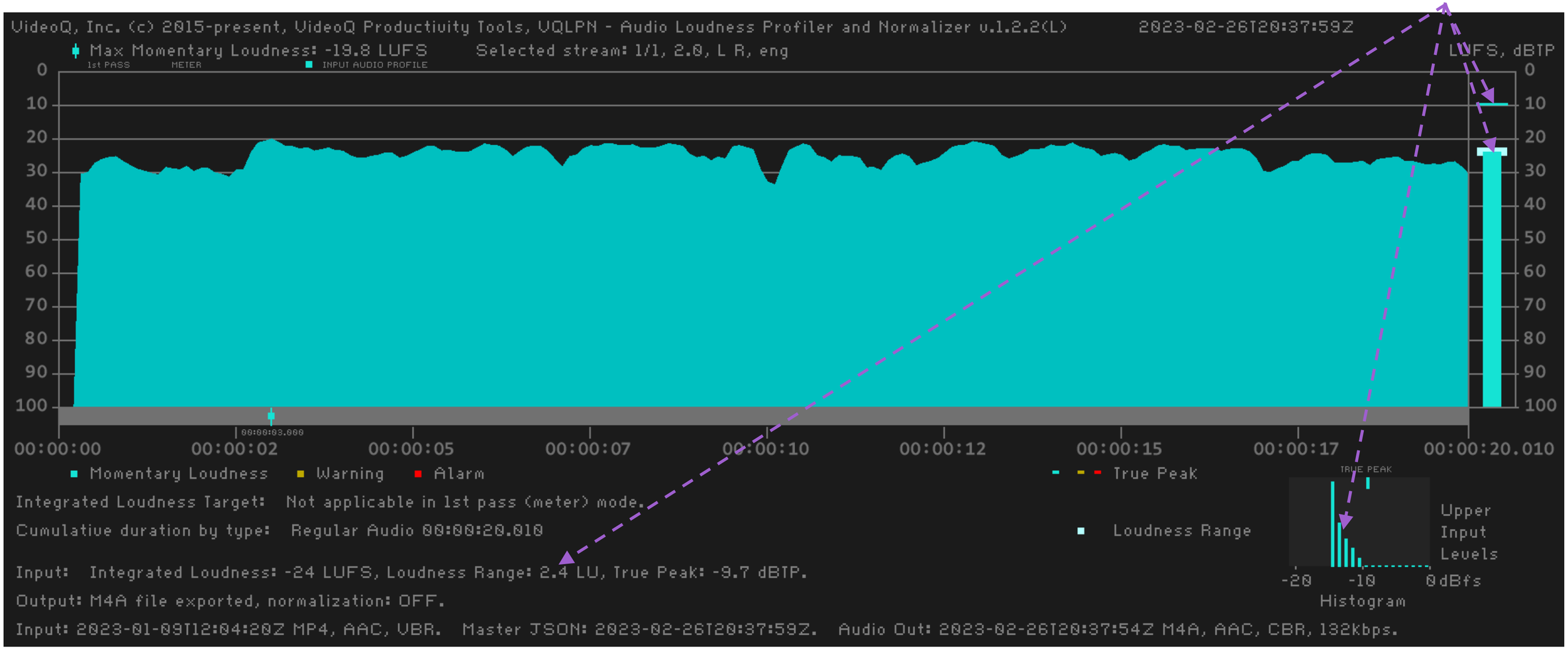
3.5 VQLPN – Audio Loudness Profiler & Normalizer

- VQLPN reads media file, containing audio stream(s), measures and optionally normalizes audio loudness.
- VQLPN supports MP4, MOV, MXF, WAV, W64, AAC, AC3, EAC3, etc., and various audio stream formats:
  - any bit depth, bit rate and sampling rate, all audio codecs supported by ffmpeg
  - multi-channel and multi-track formats: **1.0, 2.0, 5.1, 7.1**
- It measures the audio stream loudness parameters in accordance with Recommendation ITU-R **BS.1770-4** (*USA **ATSC RP A85**, EBU **R128***). Editable *.INI file stores test configuration and target parameters.
- Sorts audio segments by types (**regular audio, mute, silence, test tone, test pattern**)
- Finally, VQLPN creates detailed Report in JSON format, including **Momentary Loudness Profile** data array at 100 ms step interval
- Configurable outputs:
  - **Audio file** in the **desired format,** optionally **normalized** to the desired **Integrated Loudness** target
  - **PNG image file** showing **momentary loudness** time-line profile **plot**, **loudness** statistics **bargraph**, **upper levels histogram**, as well as other useful markers and values
- Optional stand-alone utility modules: **VQLPC** *Correlator*, **VQLPP** *Plotter*, **VQILM** *IL Meter*

CLI interface: **vqlpn** [-j jsonFilePath] [-c configFilePath] [-br bitRate_kbps] -i inFilePath [-o [outFilePath]]

3.5.1 VQLPN Plot Example 1

- Professional advert clip with very short mute segment at start, measured (checked) by VQLPN in “Meter” mode
- **Normalized audio stream** – **Integrated Loudness** is exactly equal to **-24 LUFS** ATSC standard value
- **Loudness Range 2.4 LU** is rather **low**, but for this sort of content it is **normal**
- **True Peak** value is **good**, and **Upper Levels Histogram** looks **good**

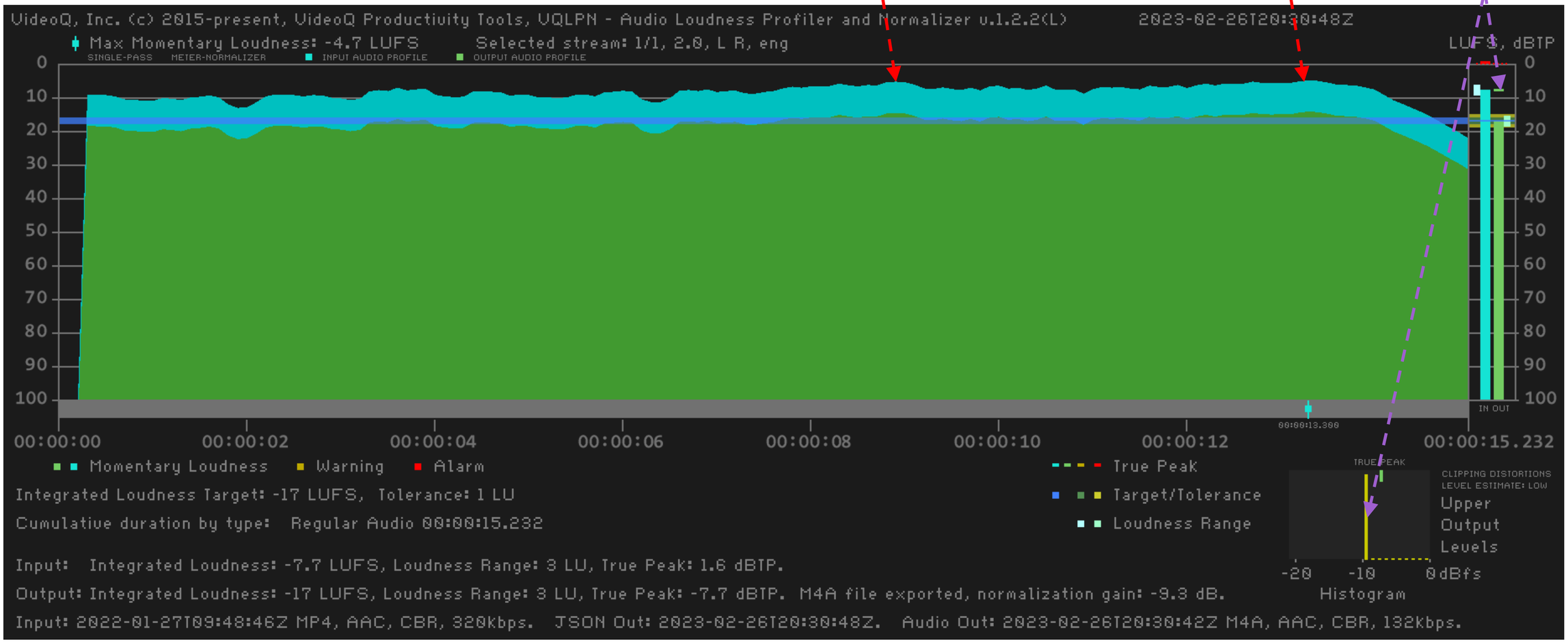


3.5.2 VQLPN Plot Example 2

- **Extremely loud** audio source **Integrated Loudness** after normalization is **exactly equal** to **-17 LUFS** (webcasting) target
- **Output True Peak** value looks **good**, but **Upper Levels Histogram** still shows **Low** level of possible **Clipping Distortions**

Conclusion: *The normalized clip is suitable for webcasting, probably with minor clipping distortions issues*

***True Peak** level now **looks good**, but **Upper Levels Histogram** still shows possible problems*



3.5.3 VQLPN Plot Example 3

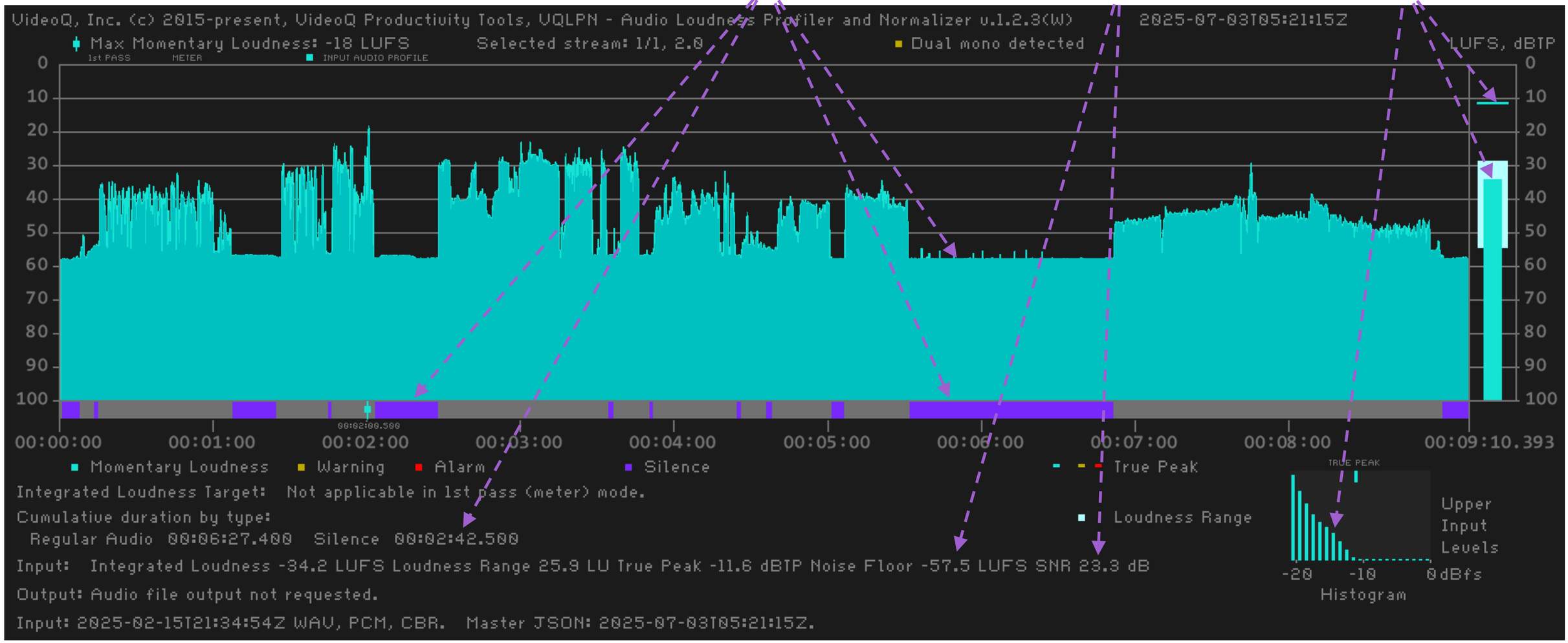
- **Scanned** 16mm film MAG track **Integrated Loudness** is **extremely low: -34.2 LUFS** (*far below all target values*)
- **True Peak** value looks rather **high** for measured low IL; this may lead to **Clipping Distortions** *after high normalization gain*

Conclusion:
*Strong **level correction** and AV timeline **editing** required*

***Silence** segments take about 30% of duration*

Noise Floor** and **SNR** values are rather **good

***IL Bar** and **Histogram** show **extremely low** levels*



3.6 VQPLA – Picture Levels Analyzer

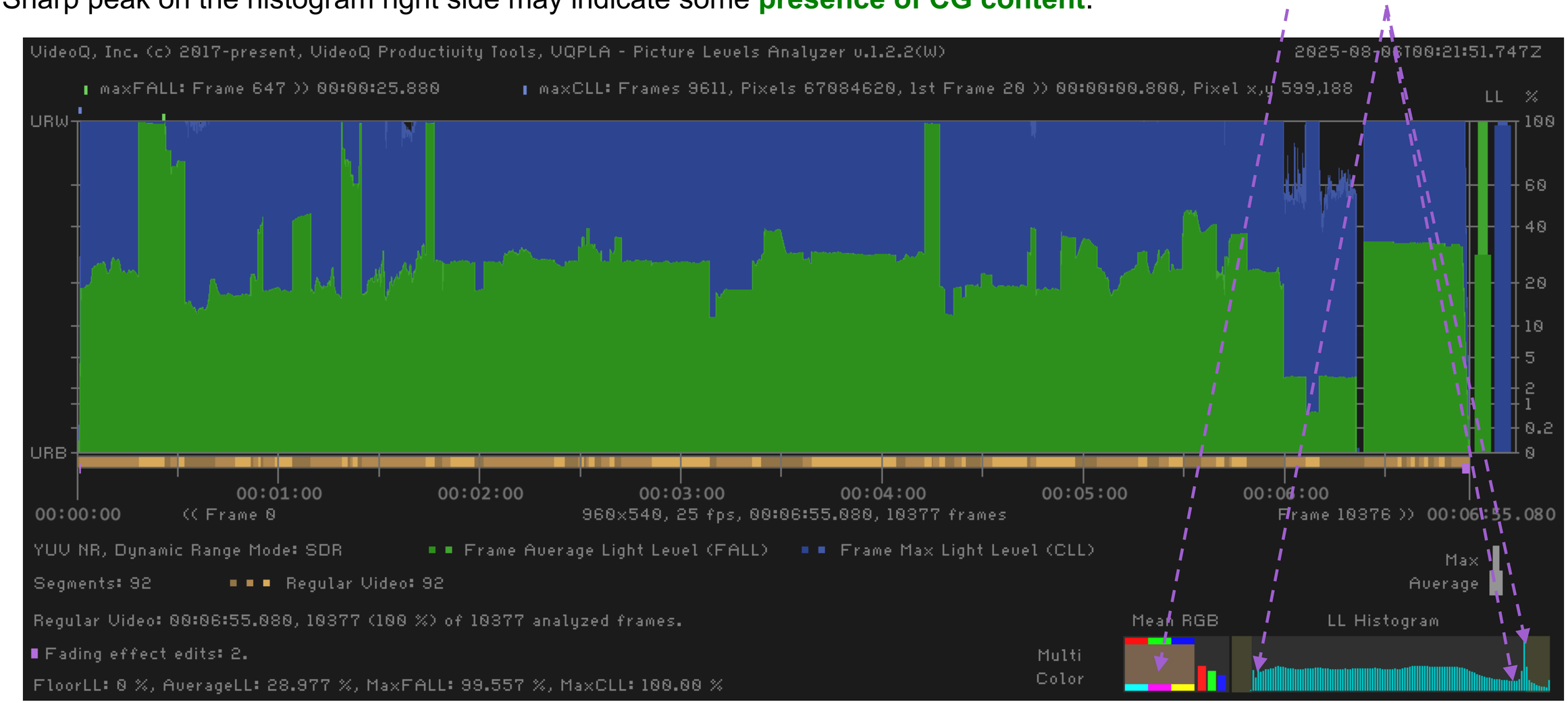
- VQPLA reads SDR/HDR media file or numbered image file sequence and measures each video frame and global parameters.
- It measures local and global Video Data Levels Statistics and Light Levels (LL) Statistics, then calculates timeline profiles of FALL and CLL levels and global LL histogram.
- VQPLA detects edit cut and cross-fade points, i.e. boundaries of video segments.
- VQPLA sorts detected video segments by types:

  Regular Video, Black, Credits On Black, Test Pattern
- Finally, VQPLA creates Report in machine-readable JSON format suitable for large databases and Plot image in PNG format suitable for human operator.

CLI interface: **vqpla** [-noplot] [-notlp] [-r float] [-DRMS] [-SRMS] -i inFilePath [-o] or [-o outFilePath]

3.6.1 VQPLA Plot Example 1

- Professional 6 min long clip containing 92 sharp edit cut segments, very short Fade-In edit at start, and Fade-Out at the end.
- **Well balanced full contrast video stream** – more or less uniform **FALL profile**, **CLL values** mostly close to 100%.
- Average Color is warm reddish-yellowish **Gray**, **Floor LL** = **0**, **LL histogram** is **spread over the valid range**.
- Sharp peak on the histogram right side may indicate some **presence of CG content**.



3.6.2 VQPLA Plot Example 2

- Digitized cultural heritage 16mm color film. Within its 6 min duration VQPLA detected 105 edit cuts and 9 fade edits.
- VQPLA added **Warning**: "*Extremely Low Average LL*". **Average LL** = **1.83%**, **MaxFALL** = **6.69%, MaxCLL** = **44.7%.**
- **Average Color** is **very dark Gray**, **Floor LL** = **0.01 %**, **LL histogram strongly biased towards low levels**.

TOC

About VideoQ

Customers & Partners

Company History

- Founded in 2005
- Formed by an Engineering Awards winning team sharing between them decades of global video technology.
- VideoQ is a renown player in calibration and benchmarking of Video Processors, Transcoders and Displays, providing tools and technologies instantly revealing artifacts, problems and deficiencies, thus raising the bar in productivity and video quality experience.
- VideoQ products and services cover all aspects of video processing and quality assurance - from visual picture quality estimation and quality control to fully automated processing, utilizing advanced VideoQ algorithms and robotic video quality analyzers, including latest UHD and HDR developments.

Operations

- Headquarters in CA, USA
- Software developers in Silicon Valley and worldwide
- Distributors and partners in several countries
- Sales & support offices in USA, UK

5. Appendix: VQPT and Media Ambit ™ Workflow