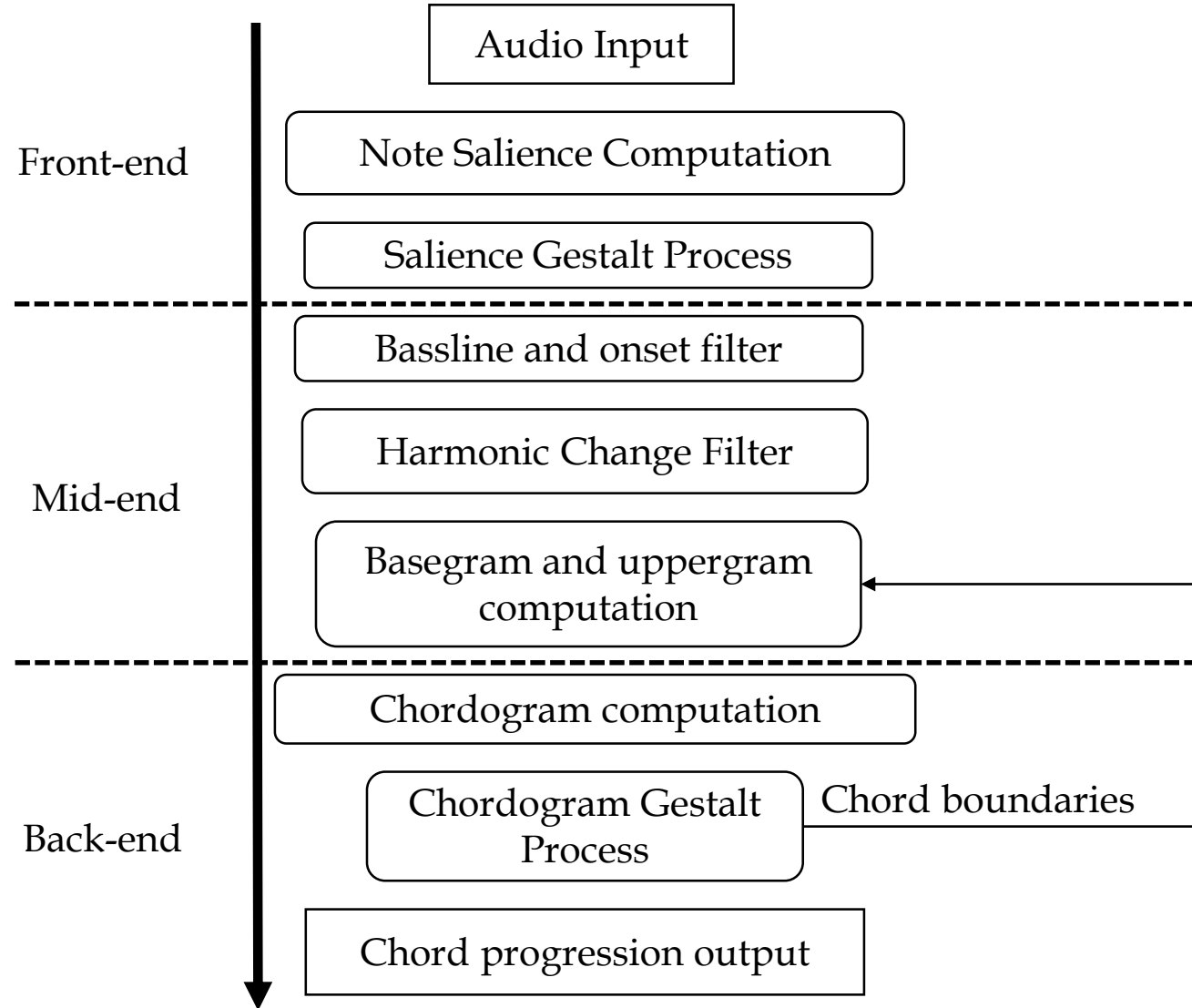# Chord Progression Recognition System

- Version 0.1
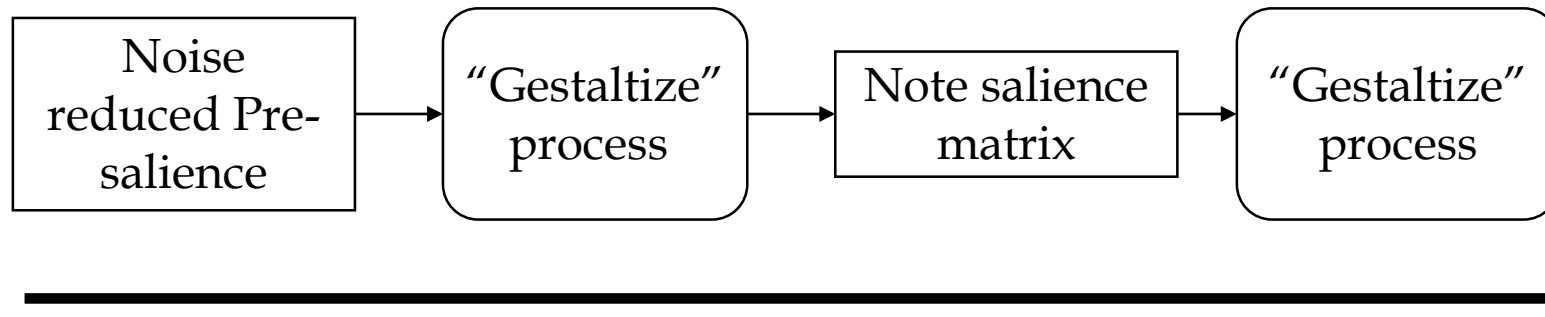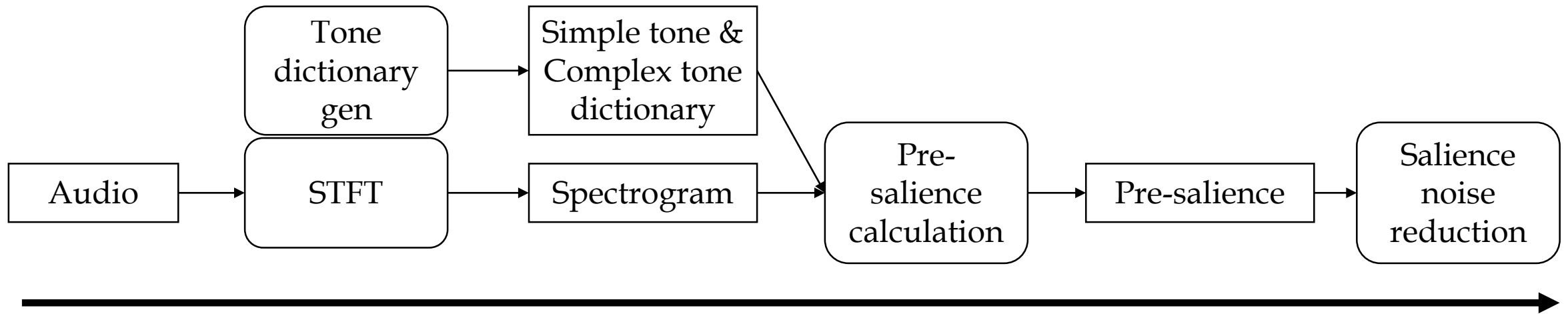
CPRS System

System Overview
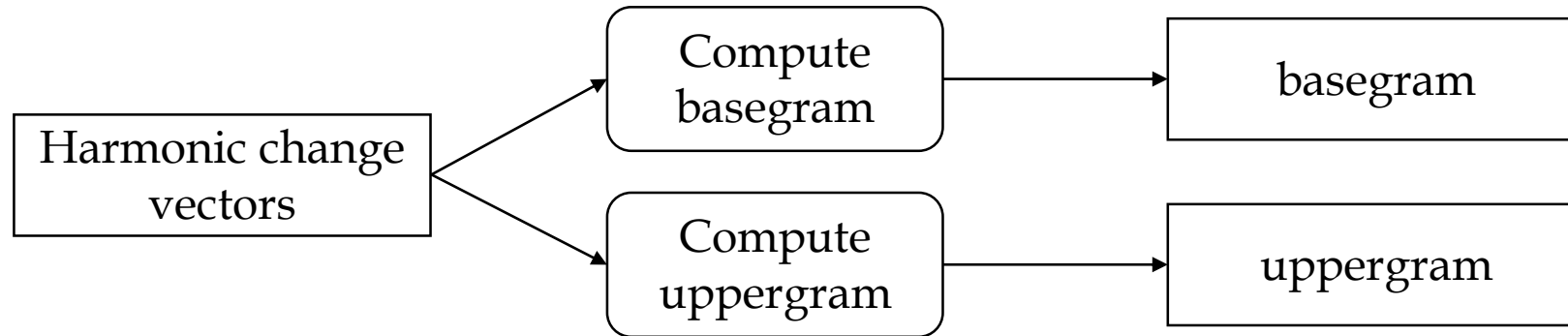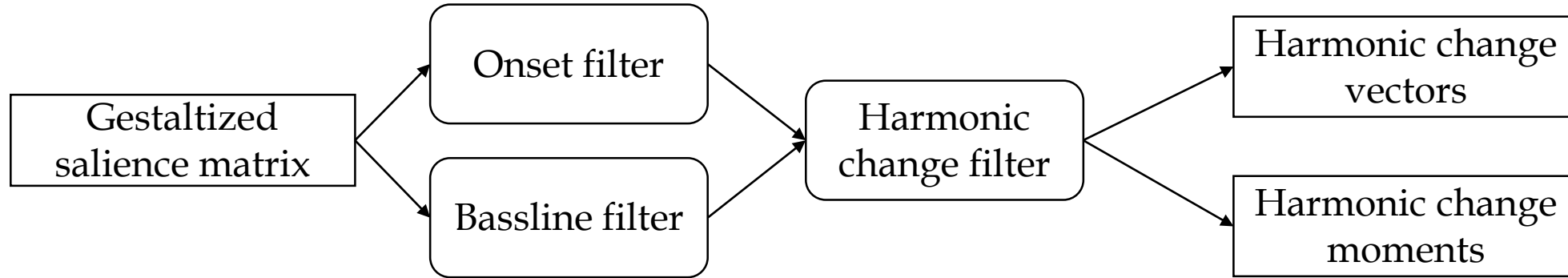
CPRS – System Overview

CPRS – Front End

CPRS – Mid End

CPRS – Back End

CPRS – Feedback

```
        ┌─────────────────┐
        │   Audio Input   │
        └─────────────────┘
                 │
                 ▼
        ╭─────────────────╮
        │    Front-end    │
        ╰─────────────────╯
                 │
                 ▼
        ╭─────────────────╮
        │     Mid-end     │◄─┐
        ╰─────────────────╯  │
                 │           │  Feedback once
                 ▼           │
        ╭─────────────────╮  │
        │    Back-end     │──┘
        ╰─────────────────╯
                 │
                 ▼
        ┌─────────────────┐
        │      Chord      │
        │   progression   │
        │     output      │
        └─────────────────┘
```
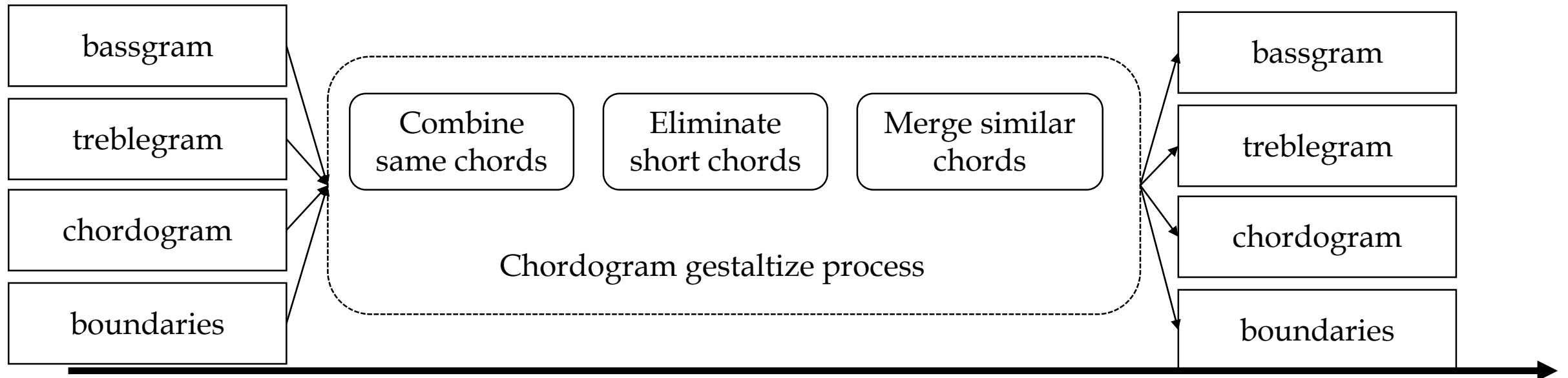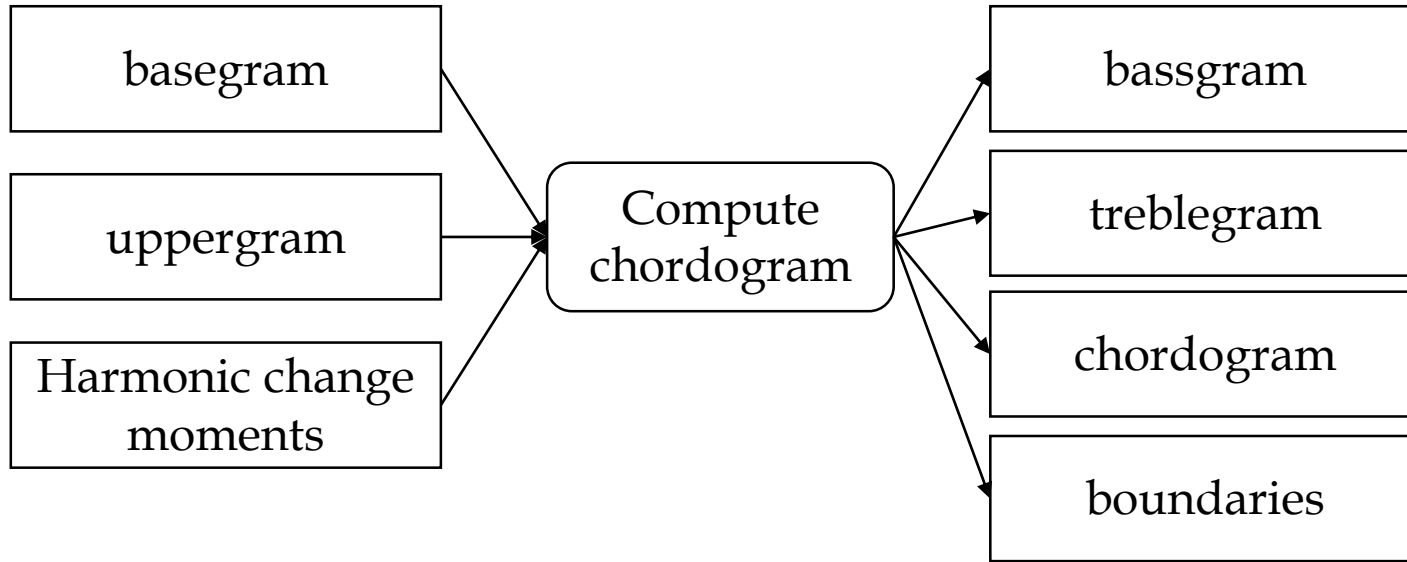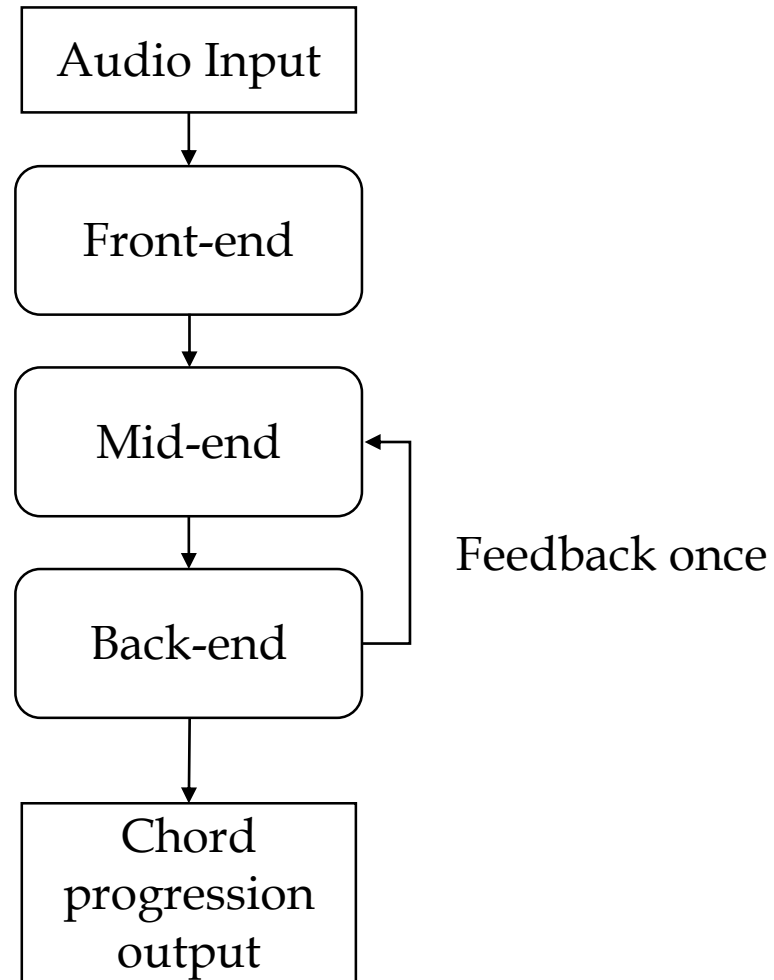
Evaluation

Chordino – my competitor
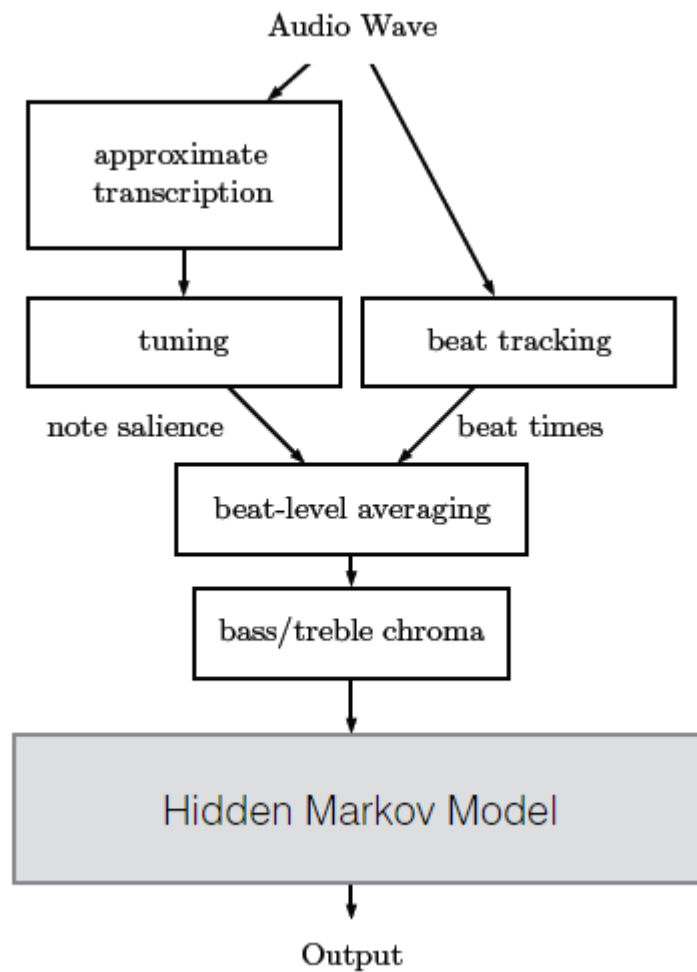
Chordino – my competitor
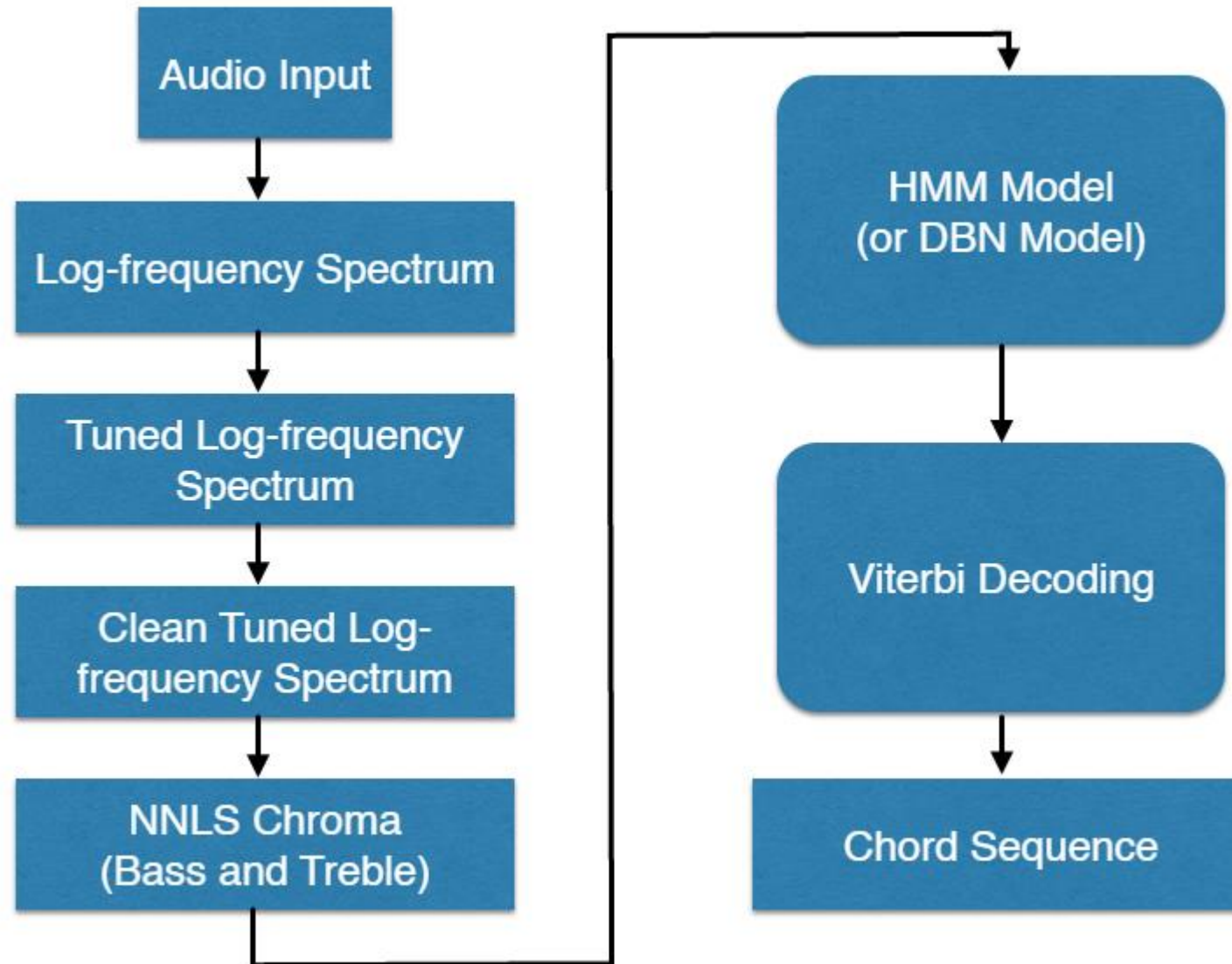
System design comparison

Key Tasks	Chordino	CPRS
Front-end	STFT, log-frequency spectrum, simple&complex tone dictionary, note salience matrix, noise reduction, tuning, nnls chroma, beat tracking	STFT, simple&complex tone dictionary, note salience matrix, noise reduction, gestalt process, onset filter, bassline filter, harmonic change filter
Mid-end	beat-level averaging, bass-chromagram and treble-chromagram	harmonic moment averaging, basegram and uppergram
Back-end	Hidden markov model, Viterbi decoding	Chordogram, chordogram gestalt process
Feedback	No	Yes

Process comparison

Key Tasks	Chordino	CPRS
spectrogram	Log-frequency spectrogram, similar to constant-Q	Only stft spectrogram
Noise reduction	Keep value larger than a certain running mean (slow)	keep peak values with certain prominence (fast)
Note salience computation	Solve an NNLS problem (deductive process, slow)	Cosine similarity between tone profiles and signal (additive process, fast)
Beat tracking	Use external beat tracking library	Beyond beat tracking, tracks harmonic change moments using bassline and onset information
Smoothing	Use inherit hmm or dbn smoothing capability, only happen in back-end	Use gestalt process, smooth in both front-end and back-end

Process comparison

Key Tasks	Chordino	CPRS
Mid-end output	Bass chromagram and treble chromagram (the bass are sometimes not the lowest pitch found in the front-end output)	Basegram and uppergram (the "base" always is the lowest pitch found in the front-end output)
Chord sequence decoding	Viterbi decoding (slow and not really necessary)	Template matching (fast and accurate because of the gestalt process in front-end)
Chord model	Triad, sixth, seventh, slash chords	Triads, sixth, seventh, sustain, slash chords…

Comparison Metrics

- bass relative correct overlap (brco) =
bass matching frames / total # of frames

- treble relative correct overlap (trco) =
treble matching frames / total # of frames

- chord correct overlap (rco) =
chord relative matching frames / total # of frames

- segmentation quality (sq) =
# of correct boundaries / total # of boundaries

Comparison Methods

- definition of correct bass matching:

Bass 0 matches nothing

other basses match the same basses pitch (note the "#" and "b")

- definition of correct treble matching:

two treble matches if their types match

Treble 'N' matches all

- definition of correct chord matching:

 if both bass and treble match, they match

- definition of correct boundary:

a detected boundary within 1/10 of the median of gt boundary-diffs

Testcases

- Randomly choose 10 songs to represent a reasonable distribution of song styles:
  - Pop, Ballad, Folk, R&B, Rock
- And to represent various kinds of music texture
  - From pure guitar accompaniment, pure piano accompaniment to full band arrangement
  - From broken chord texture to full block harmonic texture
- And to represent different keys:
 - C, E, F, F#, G, Bb, B

Comparison results on testcases

Testcases	cdbrco	cdtrco	cdsq	cdrco	cpbrco	cptrco	cpsq	cprco
1984	0.7611	0.9598	0.7632	0.7611	0.8291	0.9382	0.8222	0.8165
anjing	0.7971	0.9488	0.75	0.7923	0.8555	0.9841	0.9806	0.8555
dontbreakmyheart	0.8938	0.9501	0.9279	0.8938	0.8787	0.9706	0.9314	0.8787
feijichangde1030	0.8083	0.982	0.7717	0.8016	0.8079	0.9591	0.9038	0.792
haoting	0.6081	0.7651	0.6712	0.5882	0.8788	0.9826	0.9375	0.8788
putongpengyou	0.8378	0.8996	0.7442	0.8086	0.8227	0.9476	0.8725	0.8201
shijiemori	0.7782	0.8848	0.7813	0.7686	0.7998	0.9554	0.822	0.7968
tuihou	0.7594	0.8826	0.6667	0.7519	0.8915	0.9735	0.9785	0.8843
woruciaini	0.5678	0.7957	0.4832	0.5545	0.8196	0.959	0.7629	0.8092
xiaoxiaochong	0.7966	0.9373	0.8087	0.7748	0.7687	0.9571	0.828	0.765
On Average	0.76082	0.90058	0.73681	0.74954	0.83523	0.96272	0.88394	0.82969

Comparison results on testcases

| Testcases	cdbrco	cdtrco	cdsq	cdrco	cpbrco	cptrco	cpsq	cprco
1984	1	1	1	1	1.089	0.977	1.077	1.073
anjing	1	1	1	1	1.073	1.037	1.307	1.080
dontbreakmyheart	1	1	1	1	0.983	1.022	1.004	0.983
feijichangde1030	1	1	1	1	1.000	0.977	1.171	0.988
haoting	1	1	1	1	1.445	1.284	1.397	1.494
putongpengyou	1	1	1	1	0.982	1.053	1.172	1.014
shijiemori	1	1	1	1	1.028	1.080	1.052	1.037
tuihou	1	1	1	1	1.174	1.103	1.468	1.176
woruciaini	1	1	1	1	1.443	1.205	1.579	1.459
xiaoxiaochong	1	1	1	1	0.965	1.021	1.024	0.987
On Average	1	1	1	1	1.098	1.069	1.200	1.107

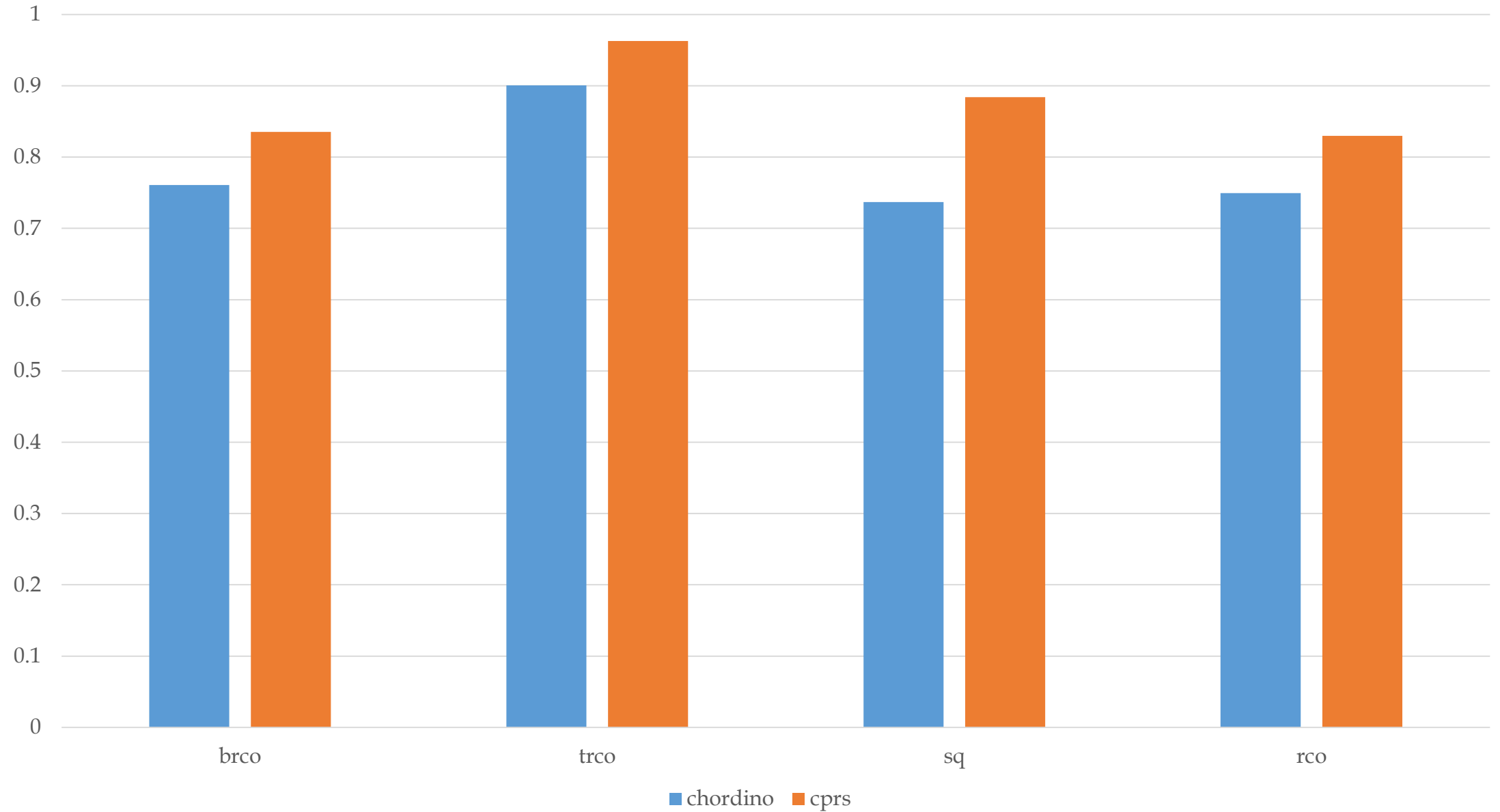Bass relative correct overlap
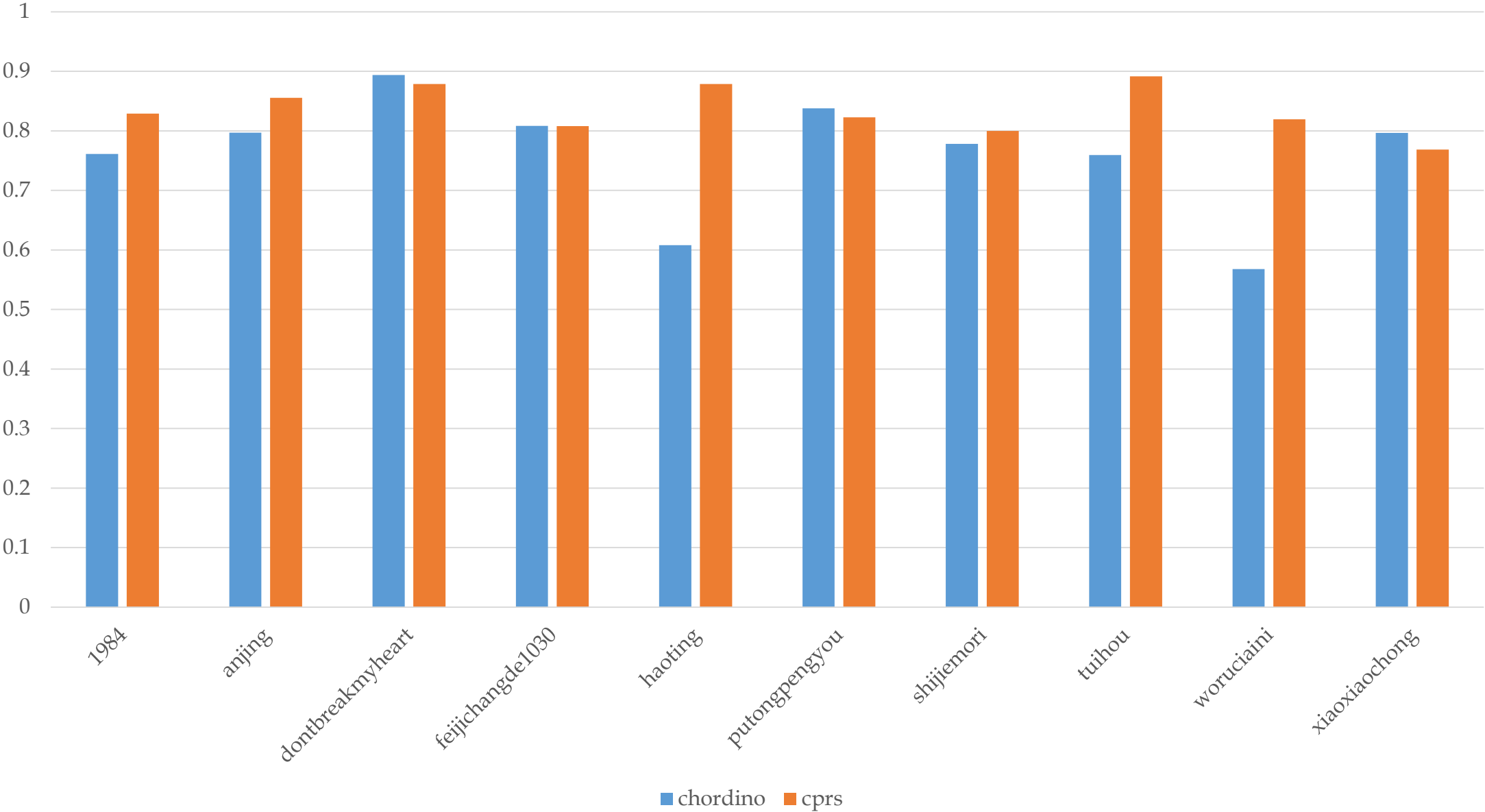
Treble relative correct overlap
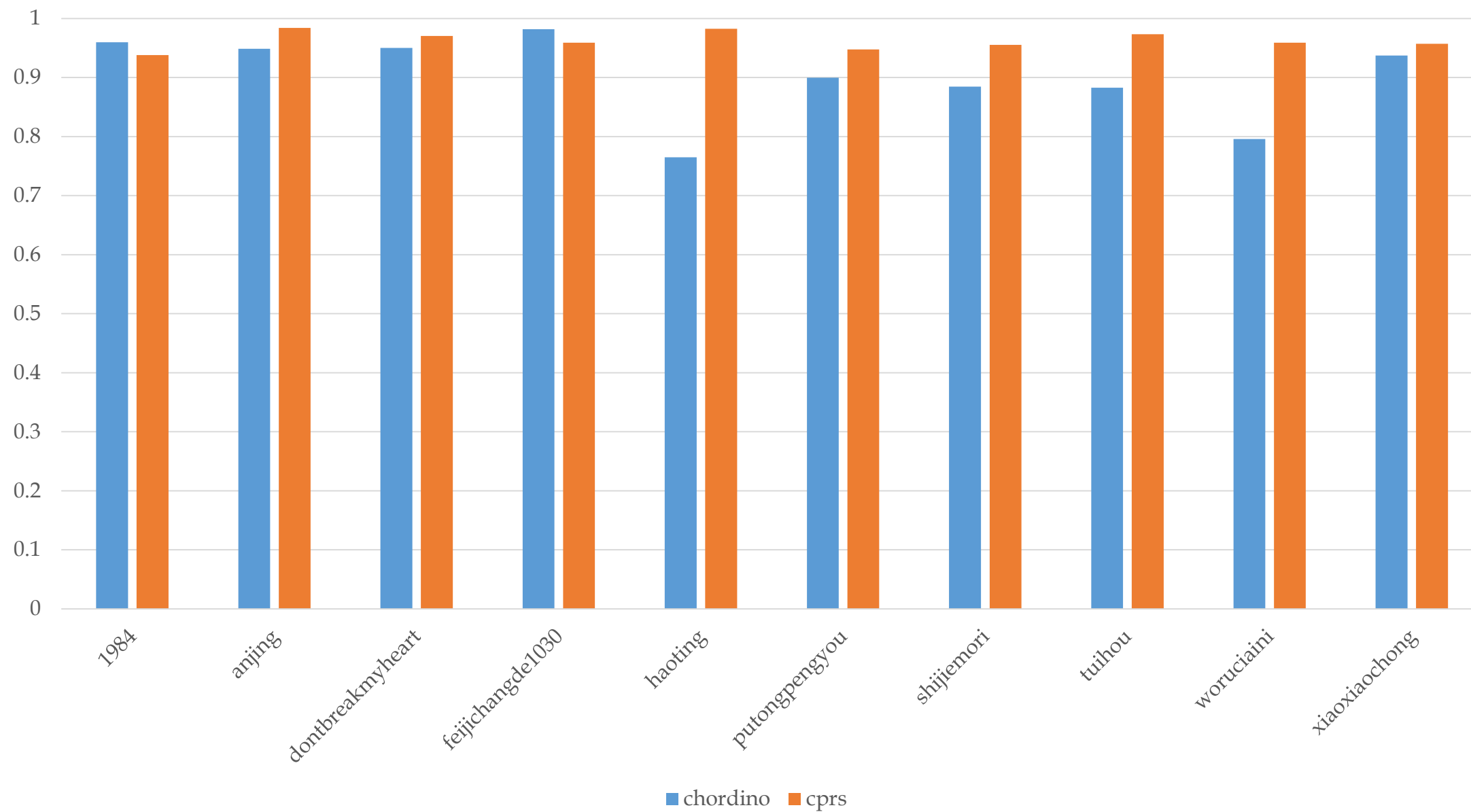
Segmentation Quality
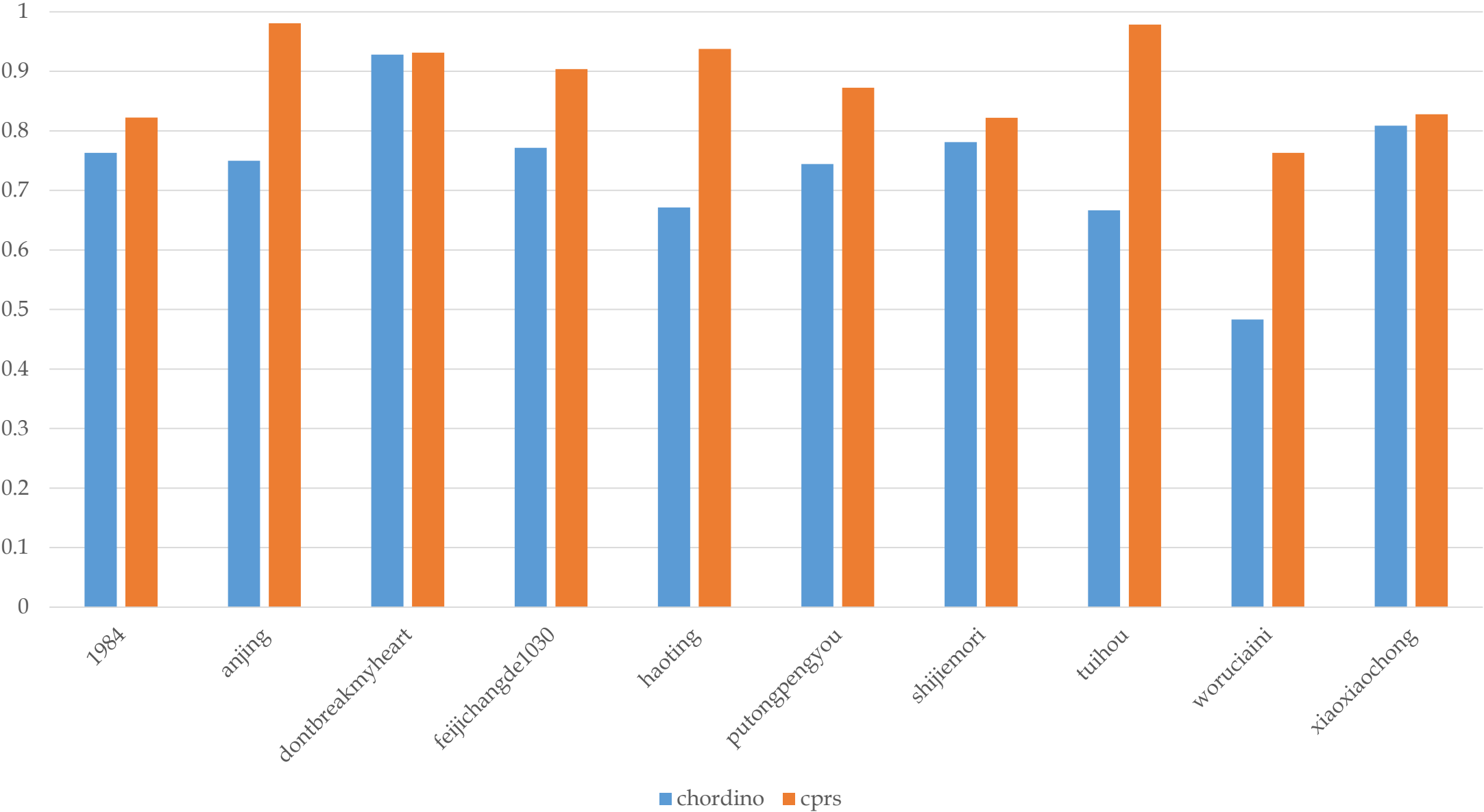
Relative Correct Overlap
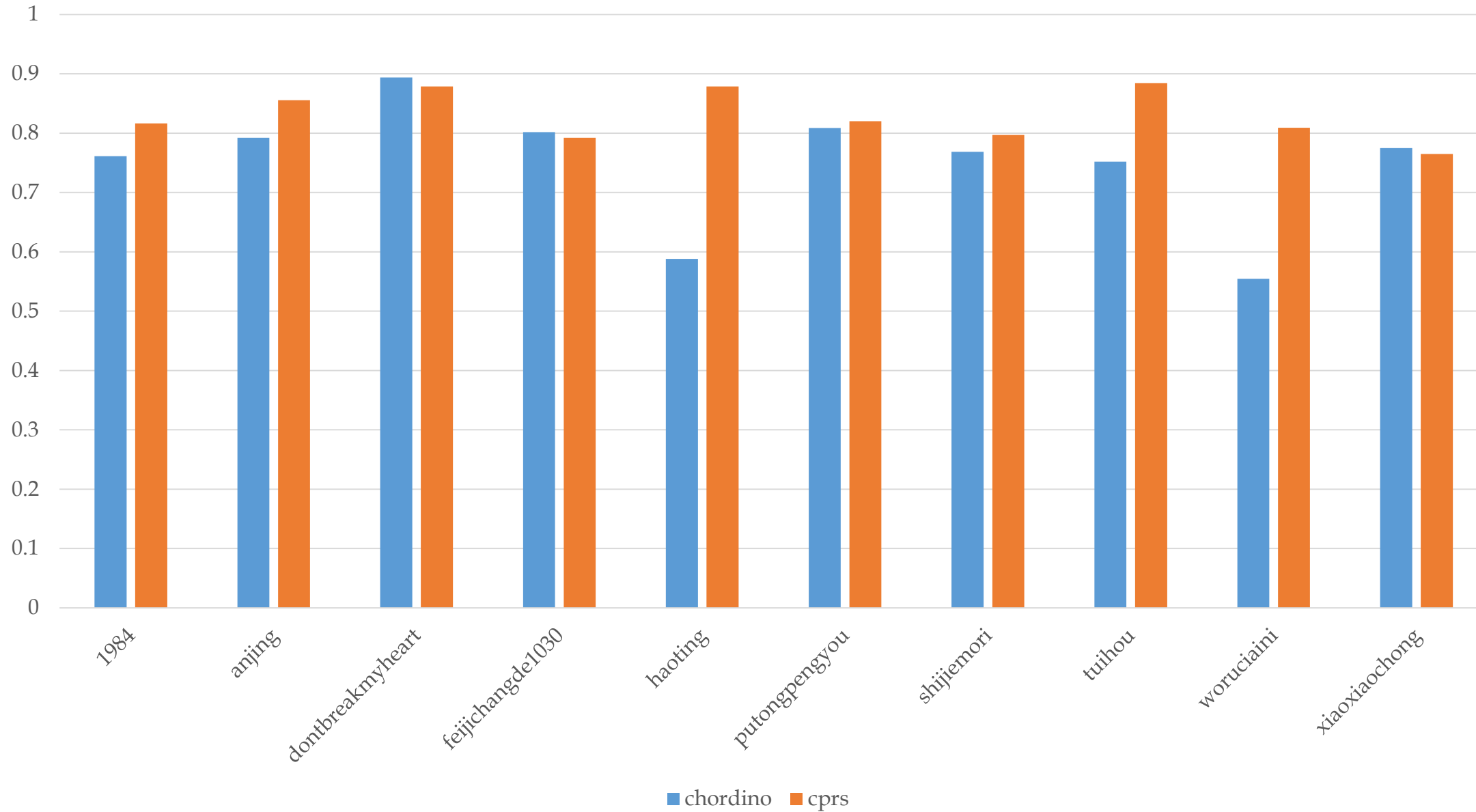
All metrics on average

Thank you!!!