UNIVERSITY OF TARTU
1632

Robust Rank Aggregation and Its Applications in Bioinformatics

Raivo Kolde

Rank Aggregation Problem

- Problem:
  - Combine preference lists into final ranking of items
- Applications:
  - Voting/elections
  - Web search
  - Information Retrieval

	D1	D2	D3	D4	D5
212489_at	10	5	938	2	2
203325_s_at	2	100	914	3	3
202310_s_at	3	119	382	9	4
211161_s_at	16	88	11841	88	8
202404_s_at	18	4	543	489	15
221729_at	22	7	150	715	53
202403_s_at	9	167	4851	385	9
202766_s_at	5	760	2913	176	91
215076_s_at	14	38	16214	244	20
202311_s_at	29	130	1038	11	6
201852_x_at	7	9	17491	748	11
212012_at	23	649	661	214	10
212013_at	15	12210	935	599	33
211980_at	13	2079	19089	349	62
201438_at	12	1842	1596	163	28
208782_at	6	323	9171	206	41

Example of RA Problem in Bioinformatics

- Goal:
  - Associate genes/proteins with diseases (based on known associations)
- Data sources used
  - Structural similarity of proteins
  - Co-occurrences in literature
  - Similarity in regulatory areas
  - Similarity of activity patterns
  - . . .

Aerts et al, Nature Biotechnology 2006

Distinctive Features

- Data sources produce noisy results
- Some data sources can be useless
- Significance of the results has to be shown
- The set of items usually genes/proteins with known number of members

Robust Rank Aggregation

- Score one item at a time
- Compare with random case
- We expect uniform distribution

	D1	D2	D3	D4	D5
212489_at	10	5	938	2	2
203325_s_at	2	100	914	3	3
202310_s_at	3	119	382	9	4
211161_s_at	16	88	11841	88	8
202404_s_at	18	4	543	489	15
221729_at	22	7	150	715	53
202403_s_at	9	167	4851	385	9
202766_s_at	5	760	2913	176	91
215076_s_at	14	38	16214	244	20
202311_s_at	29	130	1038	11	6
201852_x_at	7	9	17491	748	11
212012_at	23	649	661	214	10
212013_at	15	12210	935	599	33
211980_at	13	2079	19089	349	62
201438_at	12	1842	1596	163	28
208782_at	6	323	9171	206	41

Robust Rank Aggregation

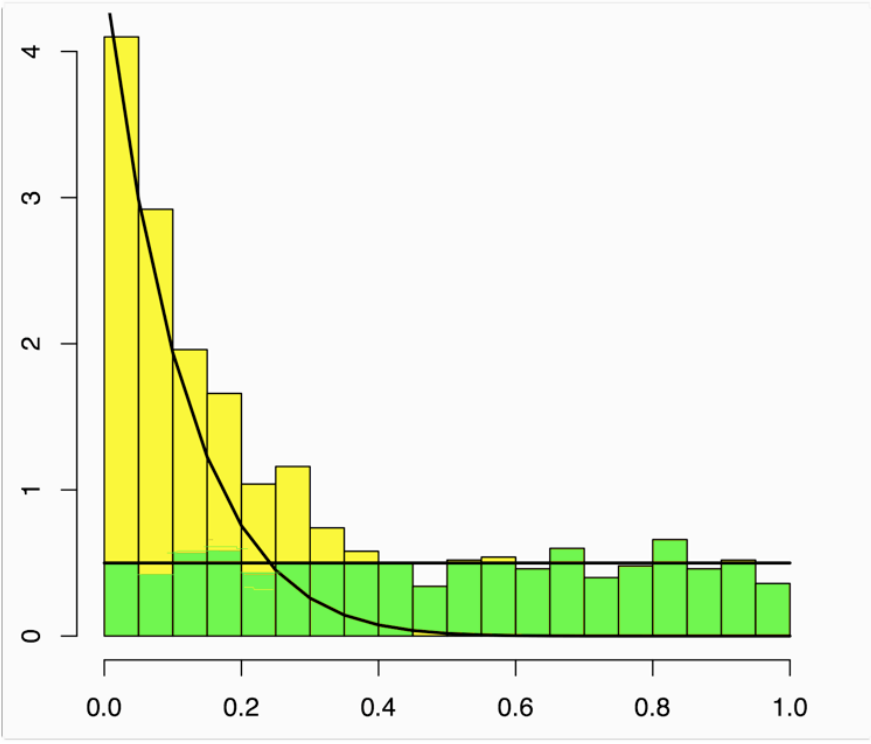
- Score one item at a time
- Compare with random case
- We expect uniform distribution

202403_s_at	9	167	4851	385	9

Robust Rank Aggregation

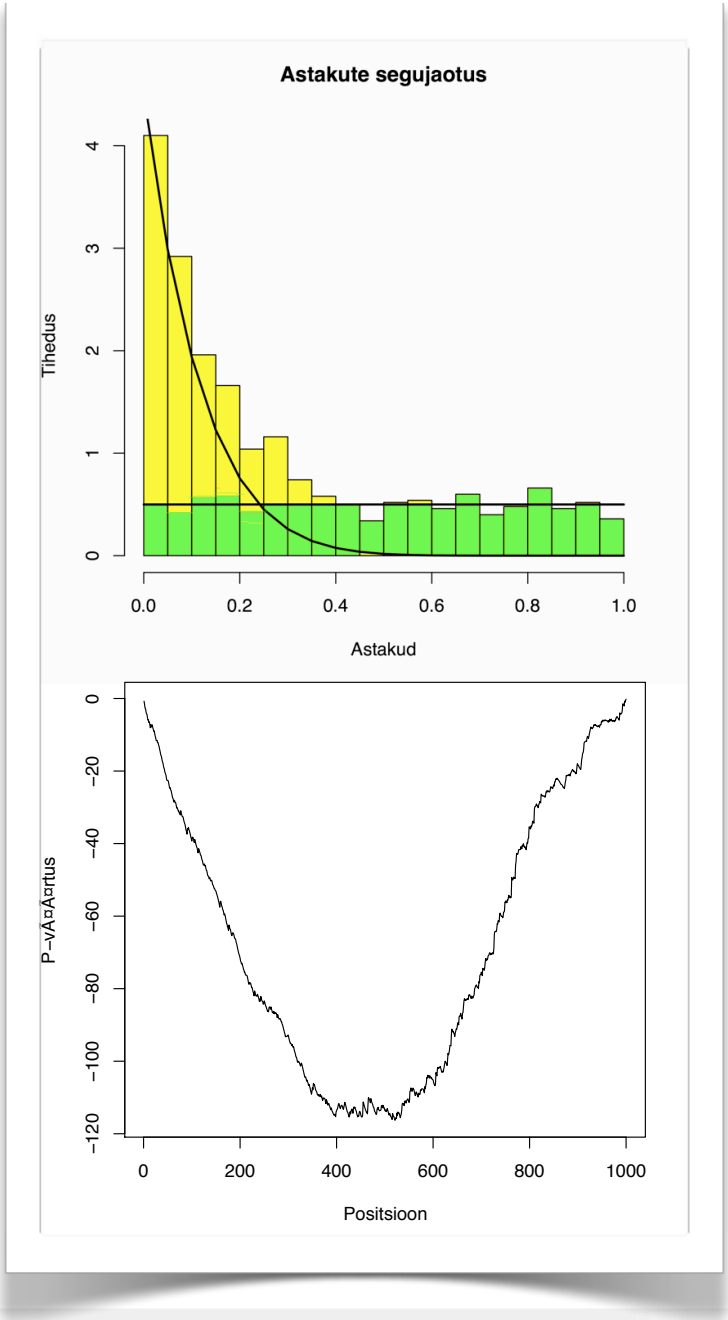
- Score one item at a time
- Compare with random case
- We expect uniform distribution

202403_s_at	9	167	4851	385	9



Algorithm

- For each item
  - Sort rank vector
  - For each sorted vector element calculate p-value
  - Use the minimal p-value as the score for item
- Sort the item based on scores
- Filter out significant results



Advantages of the Algorithm

- Insensitive to noise
- Gives significance for results
- Easy to calculate

Extension to Top-*k* Lists

- Often only top *k* ranks given
- Examples:
  - statistically sign. results
  - database queries
- Strategy:
  - Assign maximal rank to NA ranks
  - Use RRA as usual

	D1	D2	D3	D4	D5
212489_at	10	5	938	2	2
203325_s_at	2	100	914	3	3
202310_s_at	3	119	382	9	4
211161_s_at	16	88	11841	88	8
202404_s_at	18	4	543	489	15
221729_at	22	7	150	715	53
202403_s_at	9	167	4851	385	9
202766_s_at	5	760	2913	176	91
215076_s_at	14	38	16214	244	20
202311_s_at	29	130	1038	11	6
201852_x_at	7	9	17491	748	11
212012_at	23	649	661	214	10
212013_at	15	12210	935	599	33
211980_at	13	2079	19089	349	62
201438_at	12	1842	1596	163	28
208782_at	6	323	9171	206	41

Extension to Top-*k* Lists

- Often only top *k* ranks given
- Examples:
  - statistically sign. results
  - database queries
- Strategy:
  - Assign maximal rank to NA ranks
  - Use RRA as usual

	D1	D2	D3	D4	D5
212489_at	10	5	NA	2	2
203325_s_at	2	NA	NA	3	3
202310_s_at	3	NA	NA	9	4
211161_s_at	16	88	NA	88	8
202404_s_at	18	4	NA	NA	15
221729_at	22	7	NA	NA	53
202403_s_at	9	NA	NA	NA	9
202766_s_at	5	NA	NA	NA	91
215076_s_at	14	38	NA	NA	20
202311_s_at	29	NA	NA	11	6
201852_x_at	7	9	NA	NA	11
212012_at	23	NA	NA	NA	10
212013_at	15	NA	NA	NA	33
211980_at	13	NA	NA	NA	62
201438_at	12	NA	NA	NA	28
208782_at	6	NA	NA	NA	41

Extension to Top-*k* Lists

- Often only top *k* ranks given
- Examples:
  - statistically sign. results
  - database queries
- Strategy:
  - Assign maximal rank to NA ranks
  - Use RRA as usual

	D1	D2	D3	D4	D5
212489_at	10	5	25000	2	2
203325_s_at	2	25000	25000	3	3
202310_s_at	3	25000	25000	9	4
211161_s_at	16	88	25000	88	8
202404_s_at	18	4	25000	25000	15
221729_at	22	7	25000	25000	53
202403_s_at	9	25000	25000	25000	9
202766_s_at	5	25000	25000	25000	91
215076_s_at	14	38	25000	25000	20
202311_s_at	29	25000	25000	11	6
201852_x_at	7	9	25000	25000	11
212012_at	23	25000	25000	25000	10
212013_at	15	25000	25000	25000	33
211980_at	13	25000	25000	25000	62
201438_at	12	25000	25000	25000	28
208782_at	6	25000	25000	25000	41

Comparison with Common Method

- Usual method:
  - count the appearances in the top lists
  - simulate or take an arbitrary significance threshold
- Our methods advantages:
  - generalizes the common method
  - proper significance scores
  - no ties in results

Possible Applications

- Gene expression similarity search
- Network reconstruction based on gene expression
- Meta-analysis of microarray studies
- ...

Questions?