

Unraveling uncertainty in benchmarking: Methods and open-source toolkit for analyzing and visualizing challenge results

Manuel Wiesenfarth (German Cancer Research Center)

Joint work with

Annette Kopp-Schneider, Lena Maier-Hein & Annika Reinke

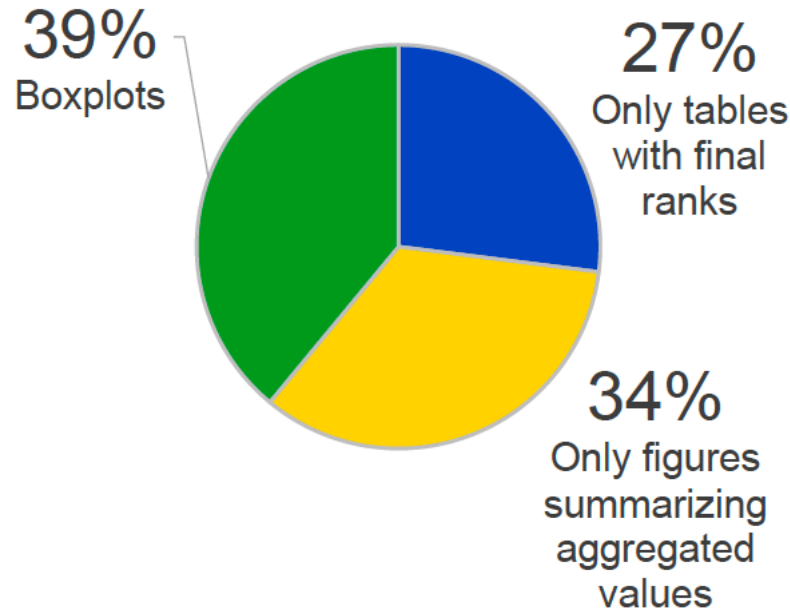
Data science seminar, Heidelberg, 11/06/2019

Background

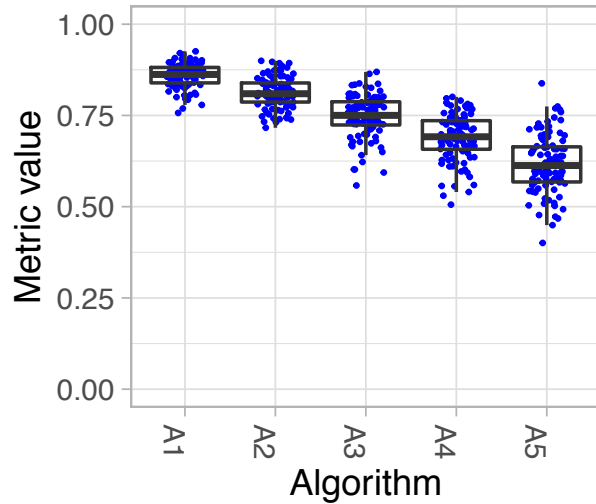
- *Grand challenges* as standard for validation of biomedical image analysis methods in a comparative manner
- Challenges compared to clinical trials
- Lack of common standards in design, analysis and reporting (Maier-Hein et al. *Nature Commun.* 2018)

Common presentation of results

In 83 challenges analyzed in Maier-Hein et al. Nature Commun. 2018:

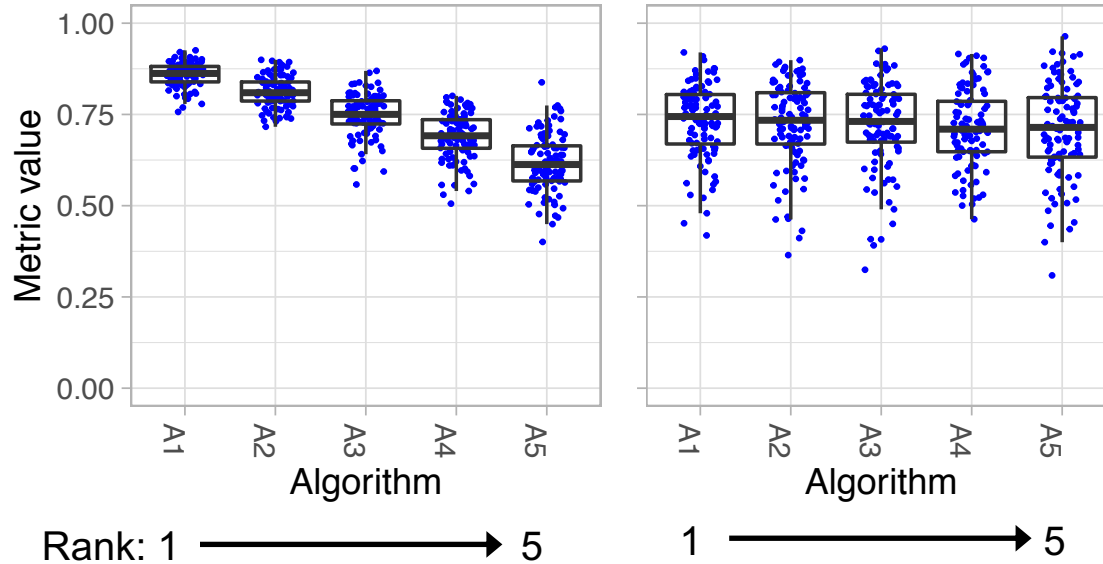


Motivation: Why ranking lists are not enough

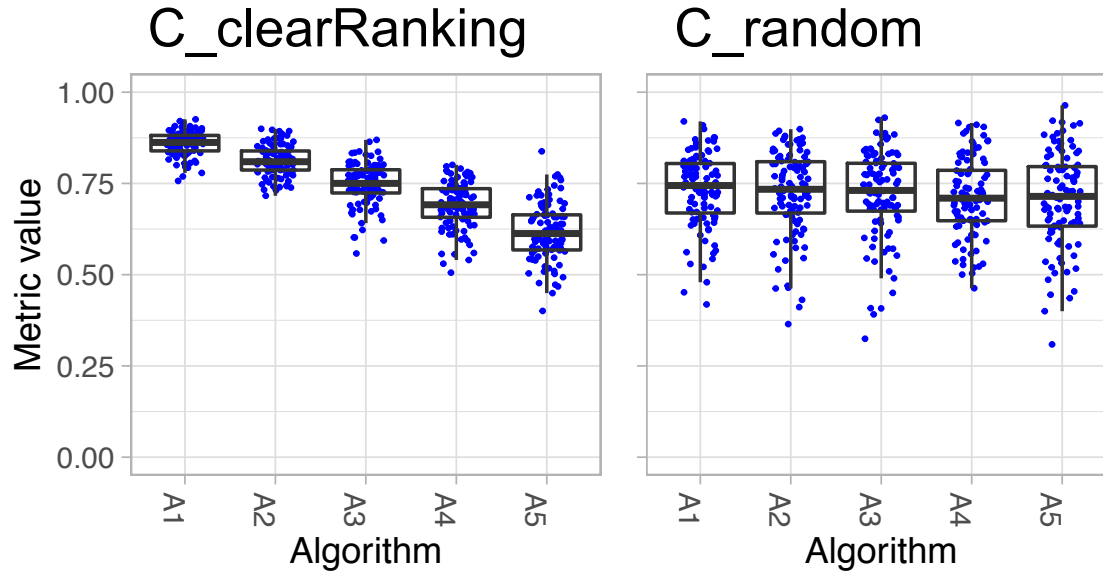


Rank: 1 \longrightarrow 5

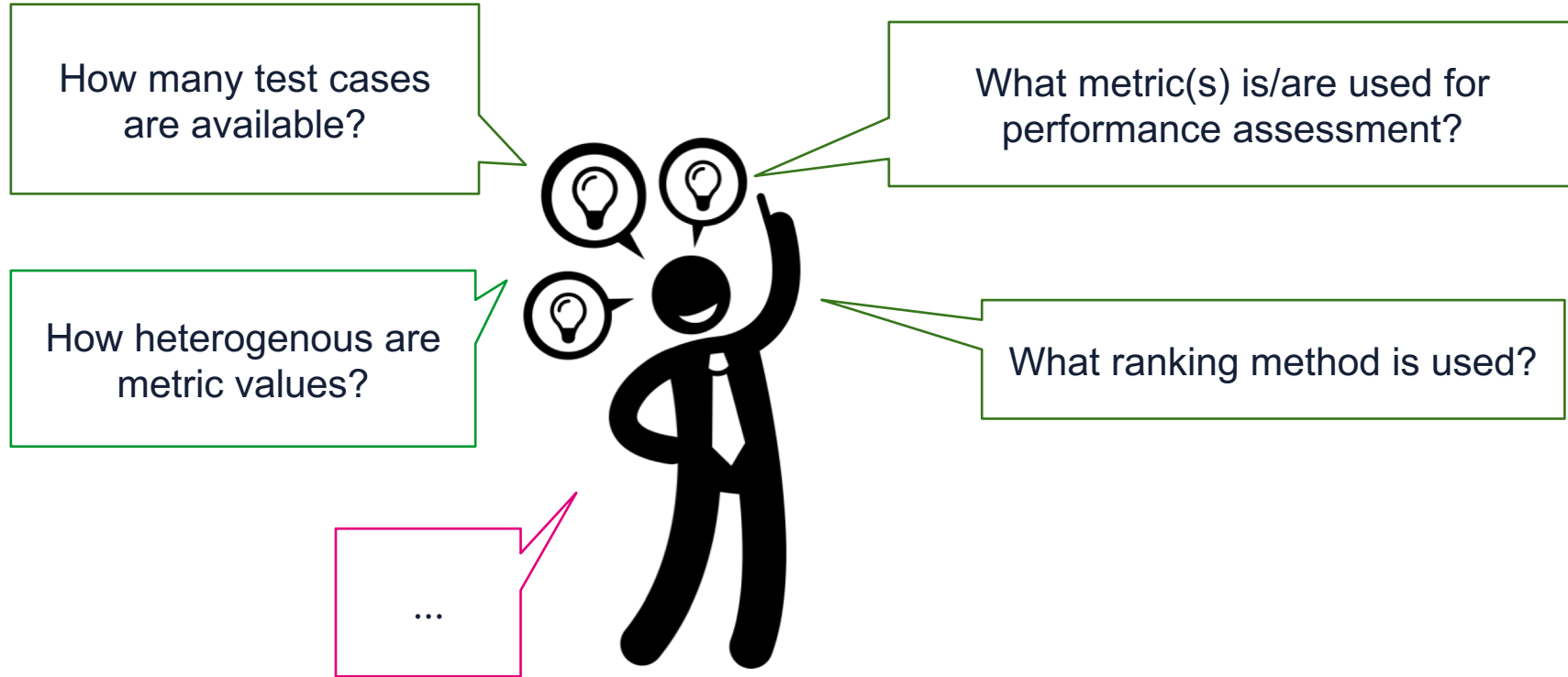
Motivation: Why ranking lists are not enough



Motivation: Why ranking lists are not enough



The ranking in a challenge may be affected by ...



Icon created by the Noun Project

Contribution

To help challenge organizers and participants gain further insights into algorithms' performances, we ...

- propose methodology for visualizing results of challenges
- provide an open-source analysis and visualization toolkit

Assessment data for a challenge analysis

Testcase_ID	Algorithm_name	Metric_value	Task_name
85	A1	0.7952	C_random
15	A4	0.6877	C_clearRanking
81	A3	0.7754	C_random
8	A5	0.6948	C_random
82	A2	0.8576	C_clearRanking
19	A2	0.5556	C_random
84	A1	0.5215	C_random

•
•
•

•
•
•

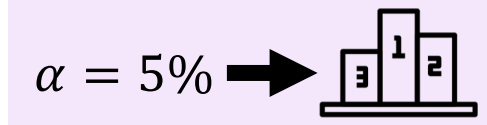
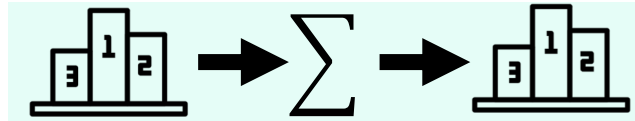
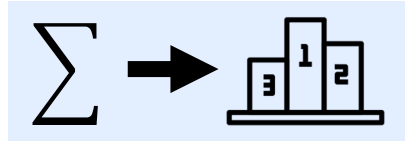
•
•
•

•
•
•

Ranking methods

Common methods:

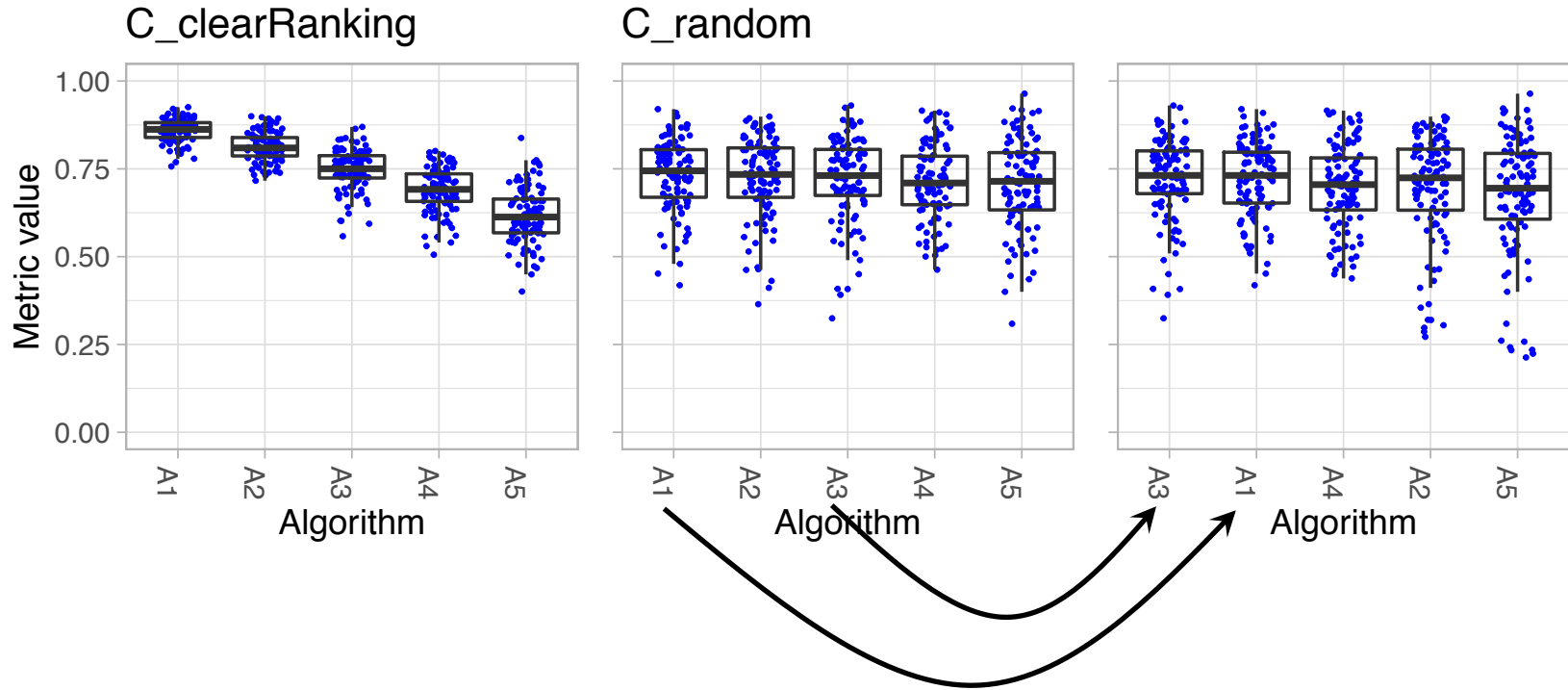
- Aggregate-then-rank
- Rank-then-aggregate
- Test based procedures



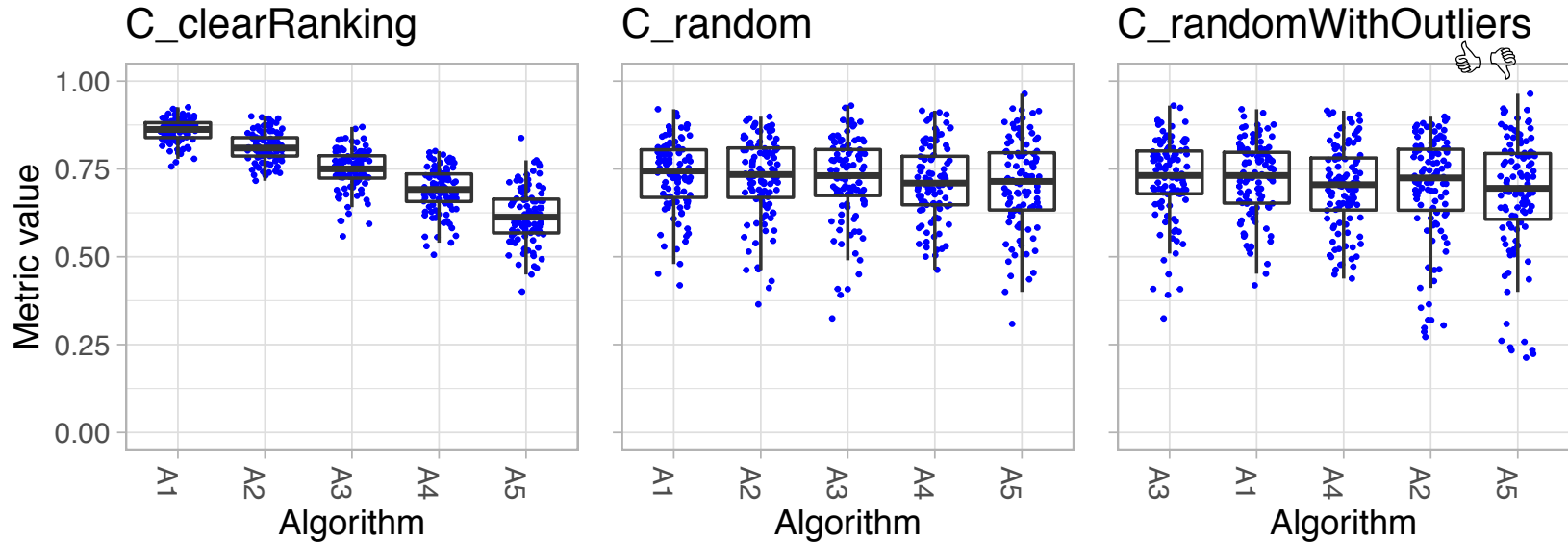
Different strategies to obtain ranking may lead to different winners

Visualization of raw assessment data: Understand distribution of metric values

Dot- and boxplot: Distribution of metric values

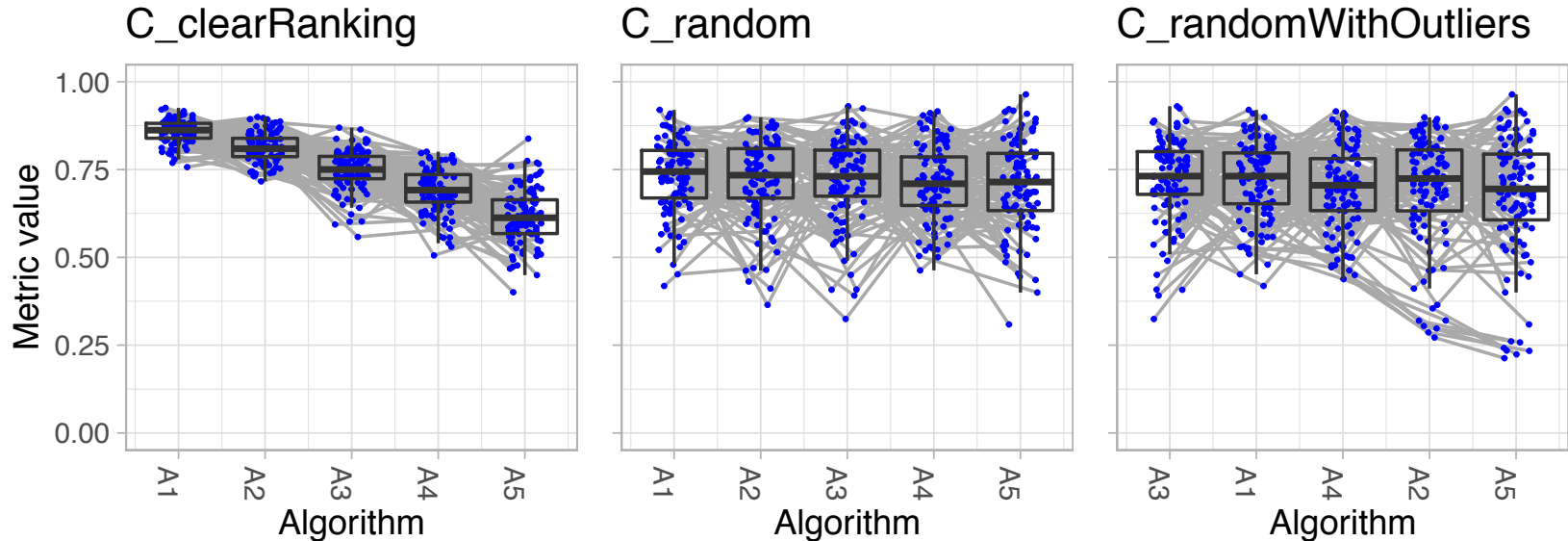


Dot- and boxplot: Distribution of metric values



+ Helps identify implausible values / inconsistencies in the data

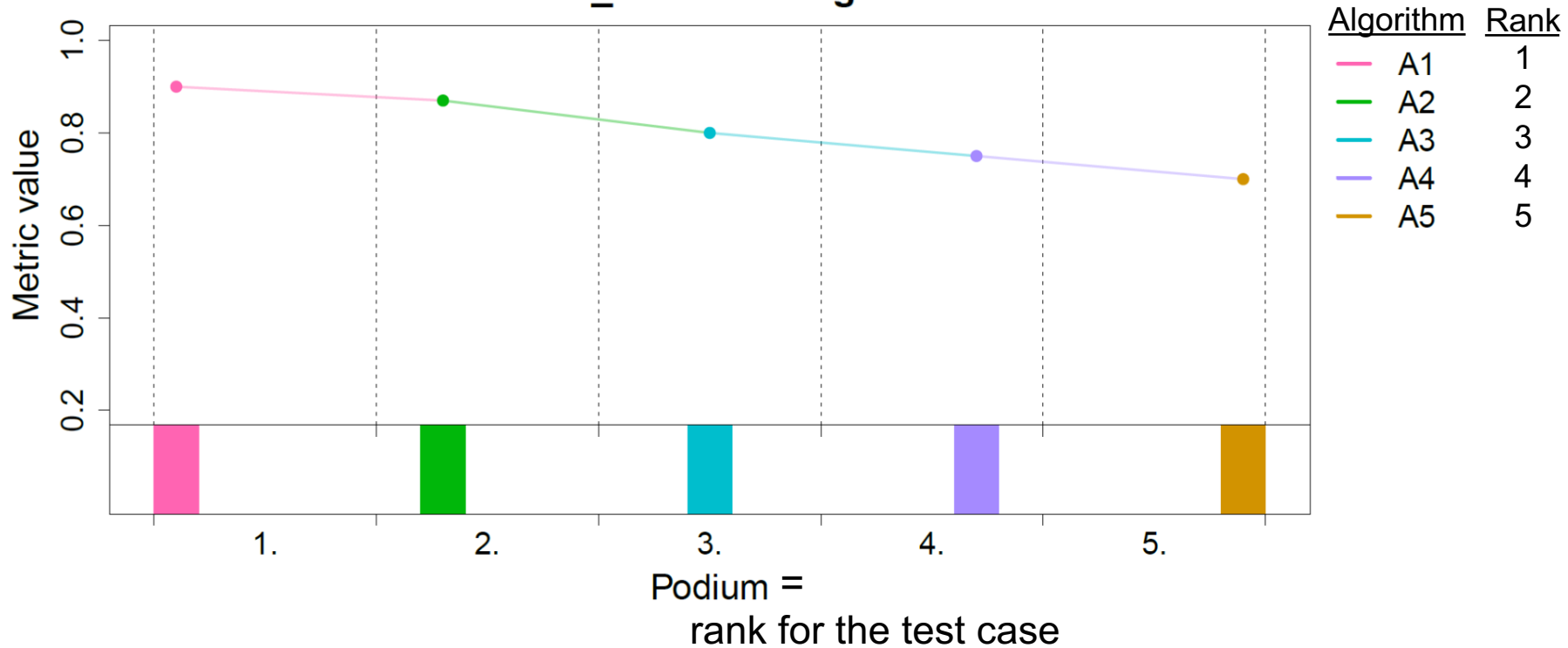
Dot- and boxplot: Distribution of metric values



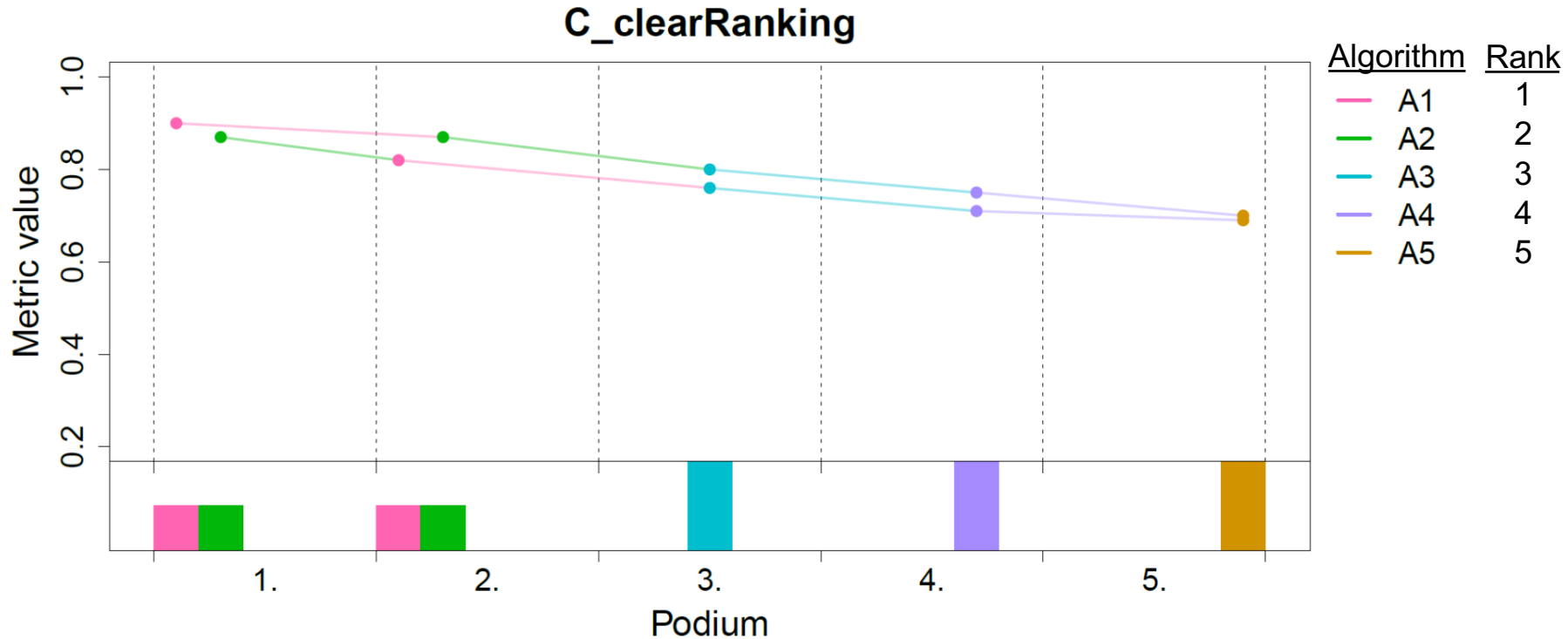
- + Helps identify implausible values / inconsistencies in the data
- Connection of metric values from the same test case?

Podium plot: Combined view of metric values and ranking

C_clearRanking

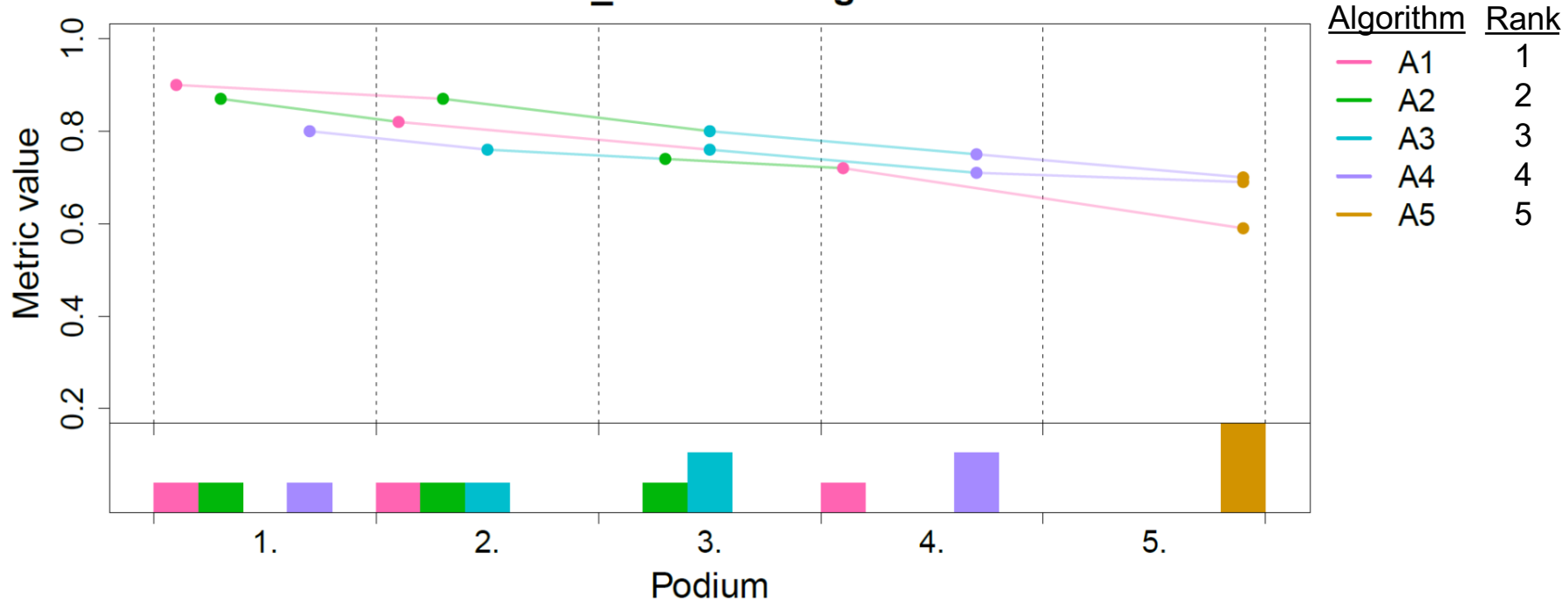


Podium plot: Combined view of metric values and ranking

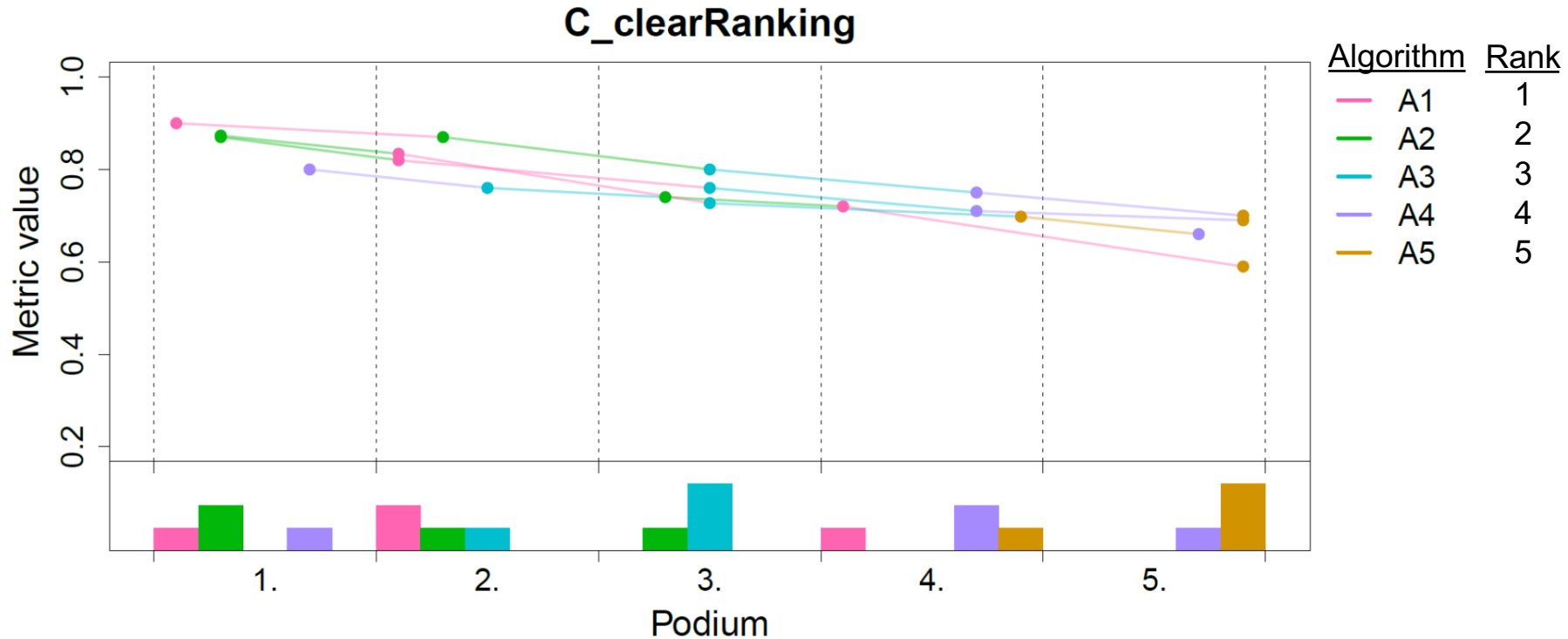


Podium plot: Combined view of metric values and ranking

C_clearRanking

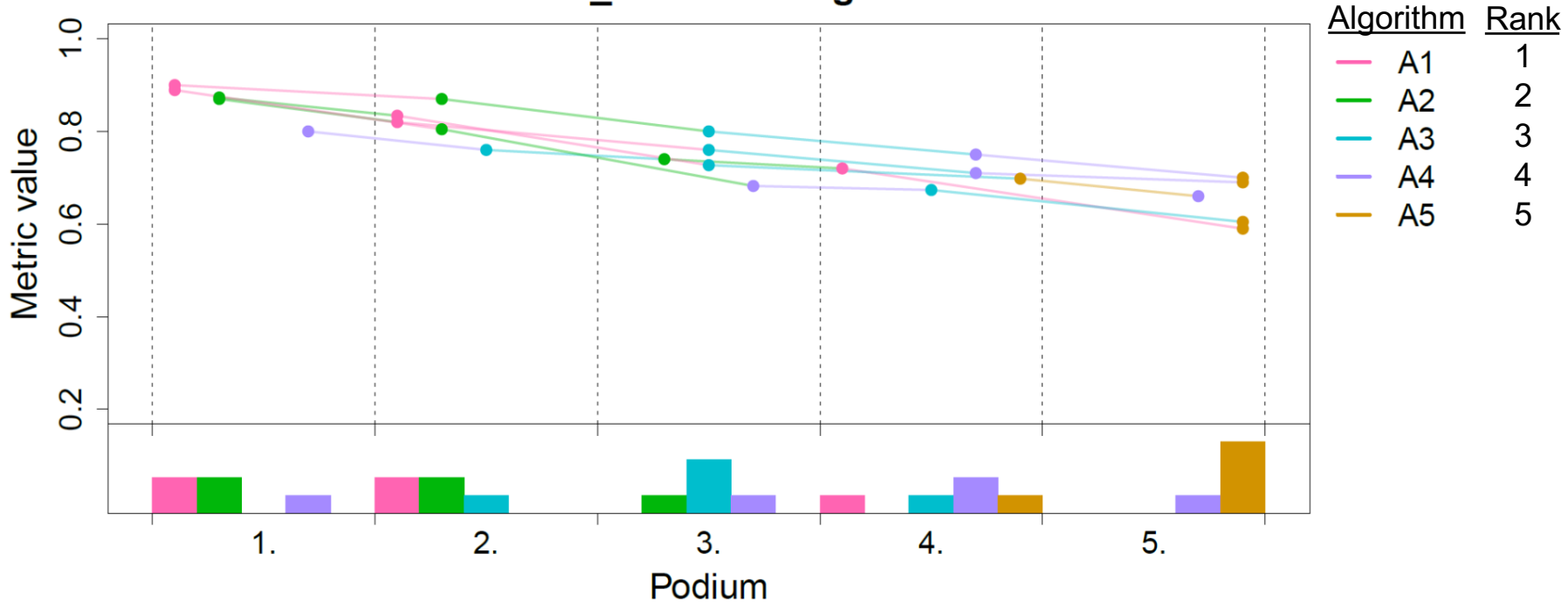


Podium plot: Combined view of metric values and ranking



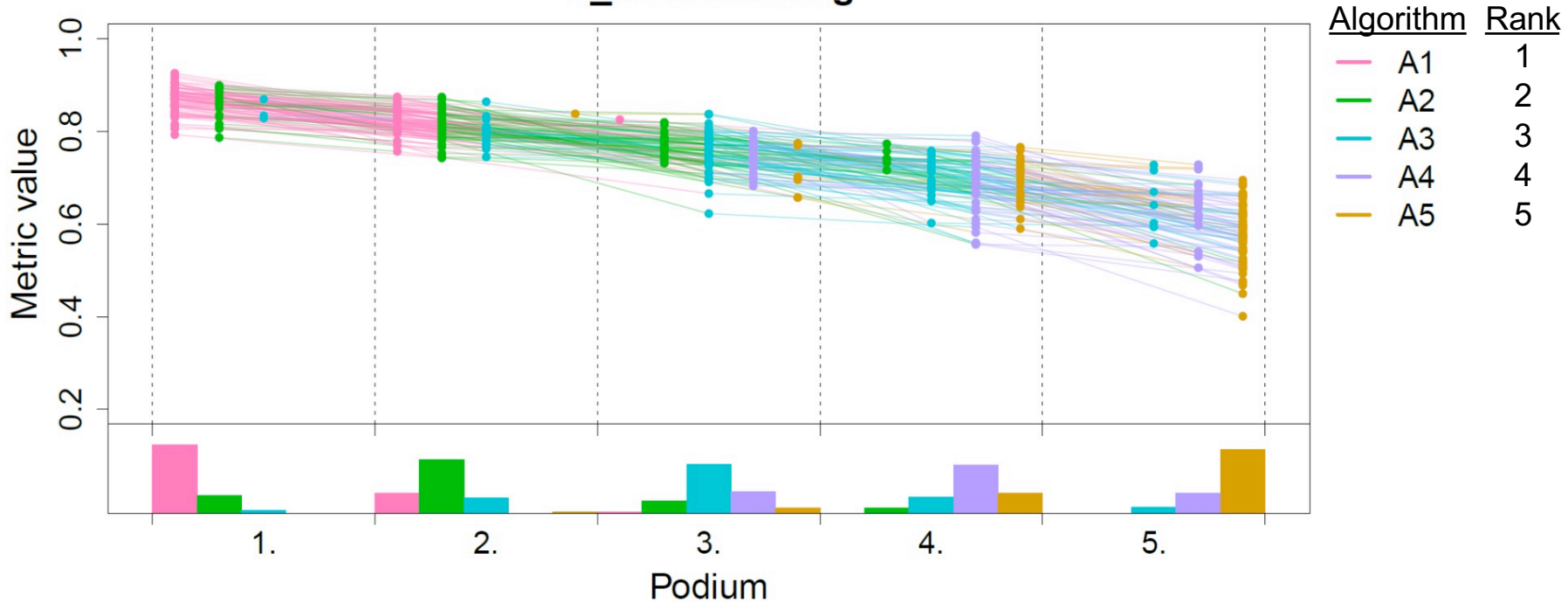
Podium plot: Combined view of metric values and ranking

C_clearRanking

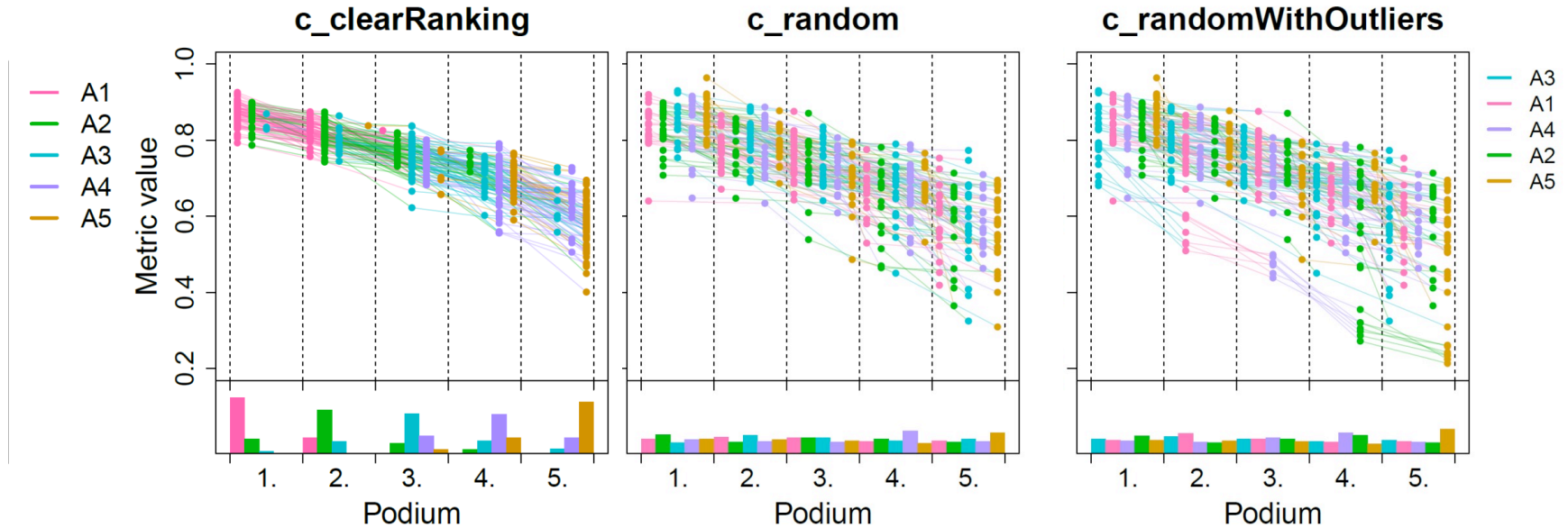


Podium plot: Combined view of metric values and ranking

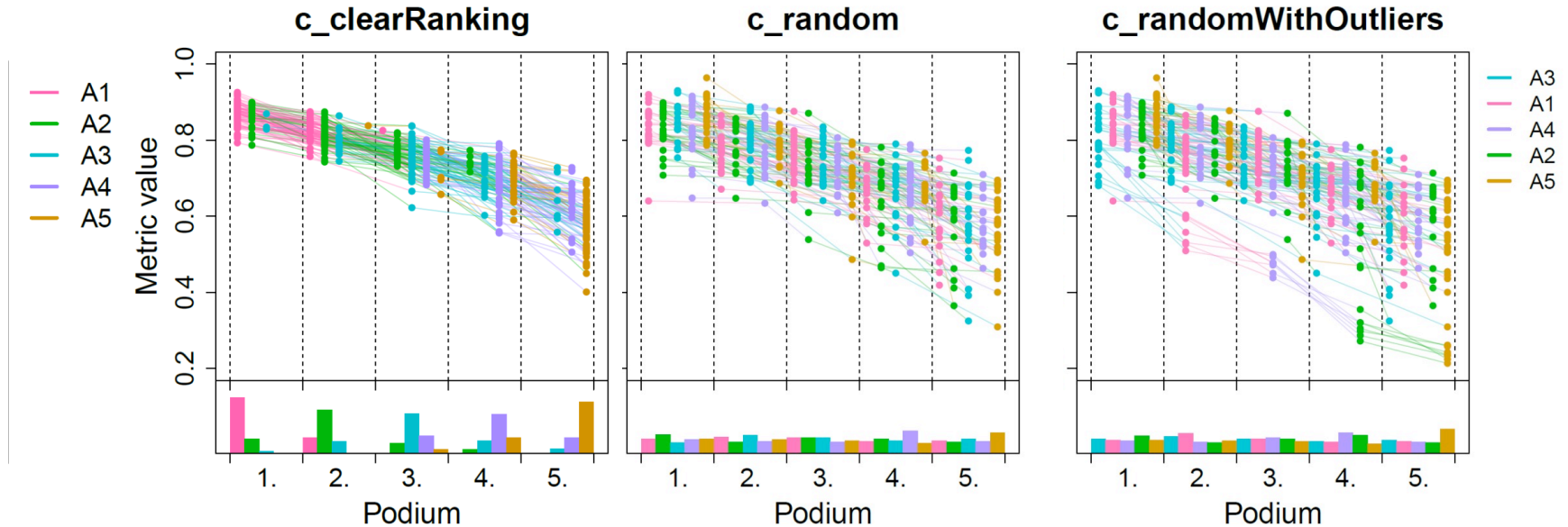
C_clearRanking



Podium plot: Combined view of metric values and ranking



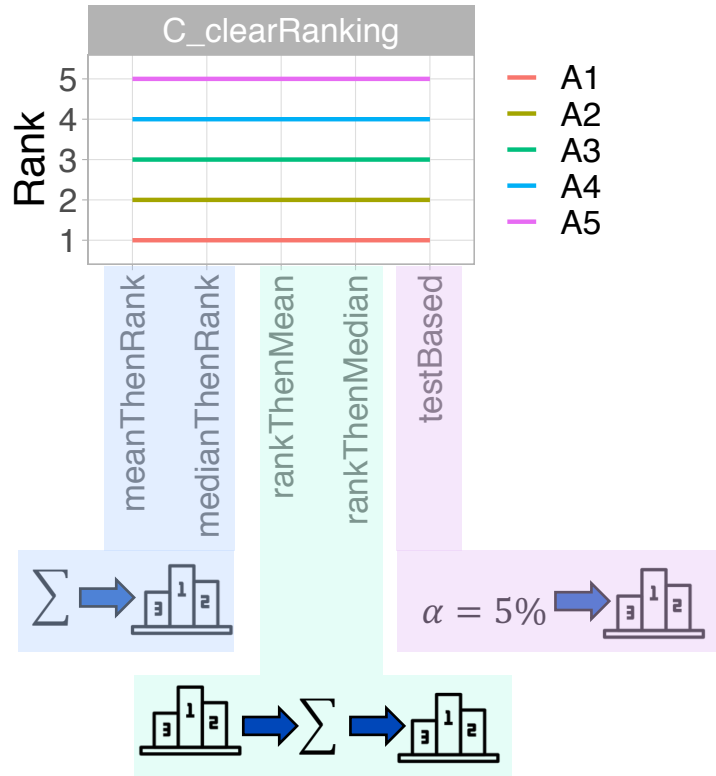
Podium plot: Combined view of metric values and ranking



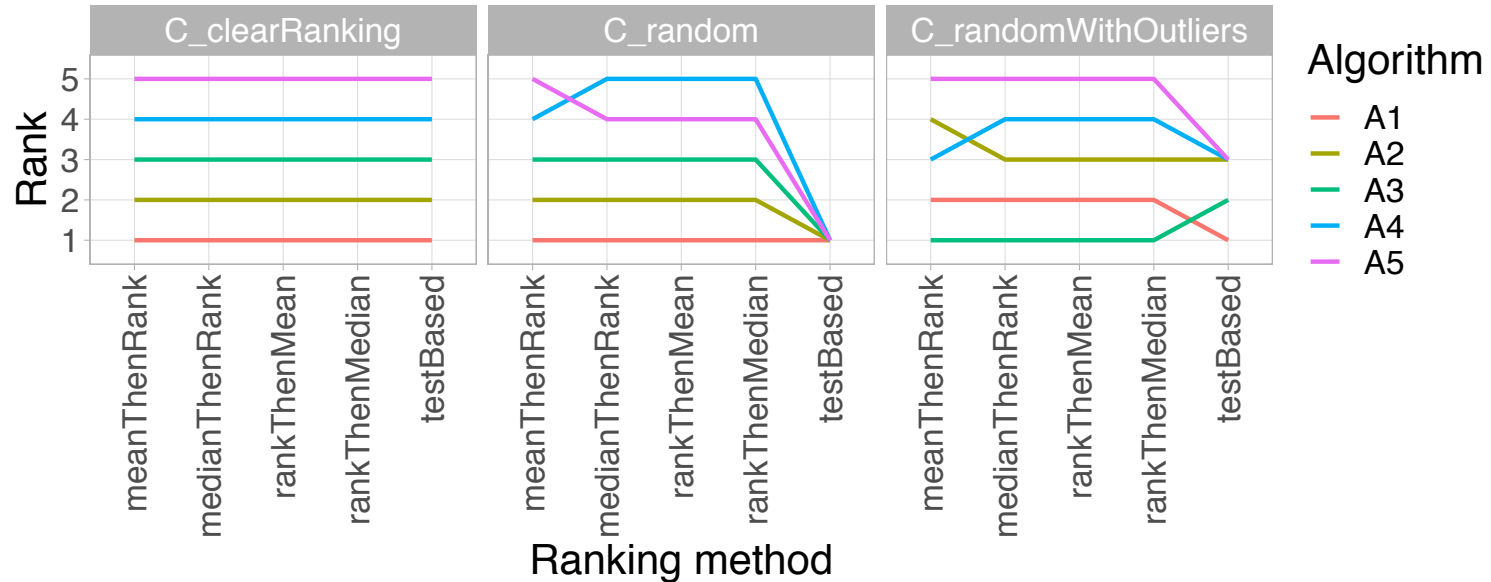
- + Connects metric values from the same test case
- + Allows identification of influential test cases
- Relatively complex

Ranking robustness and uncertainty

Line plot: Ranking robustness with regard to ranking method



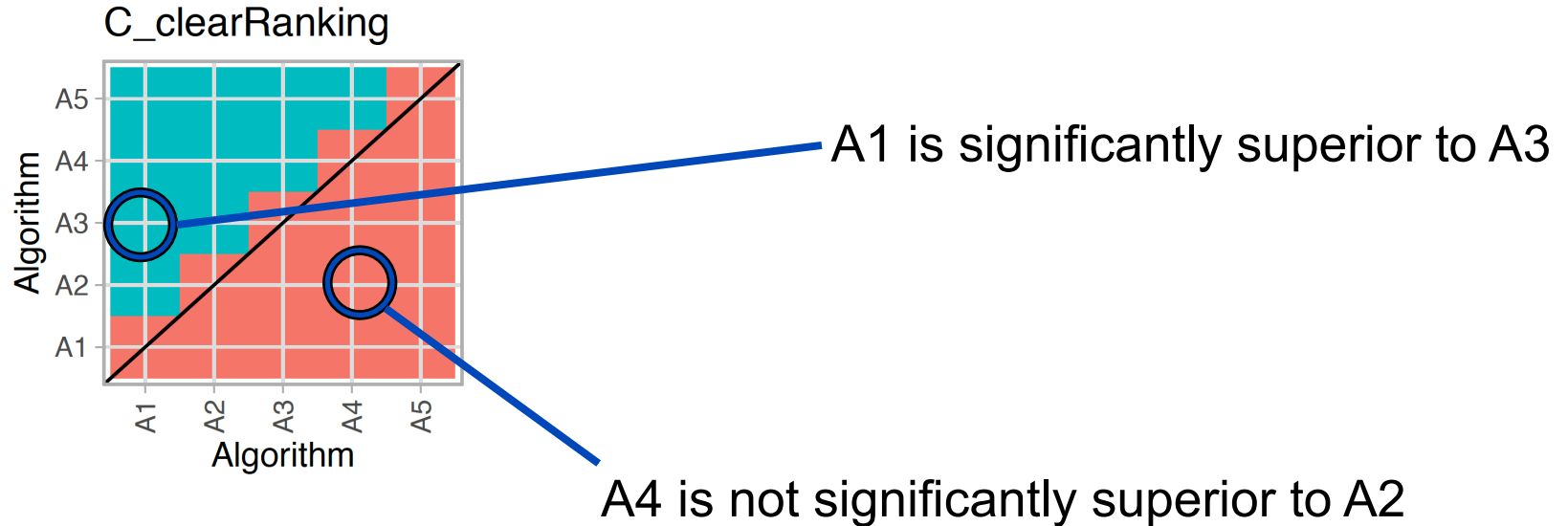
Line plot: Ranking robustness with regard to ranking method



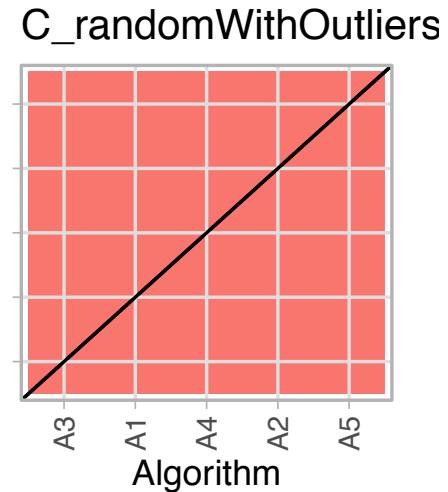
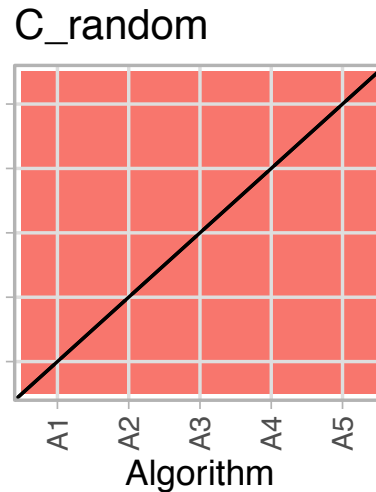
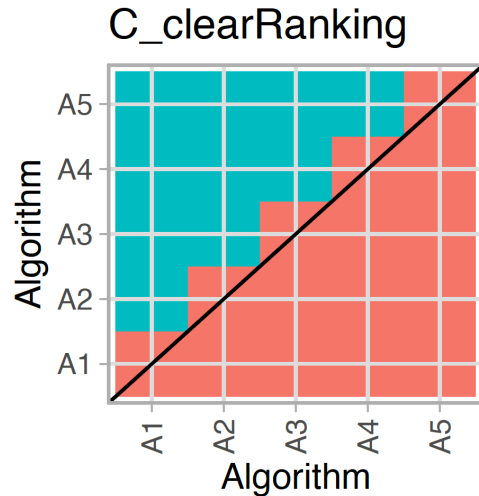
Can be modified to assess robustness with regard to metric, across tasks, ...

Significance map: Ranking uncertainty using hypothesis tests

Test cases can be considered as a finite sample from a larger population



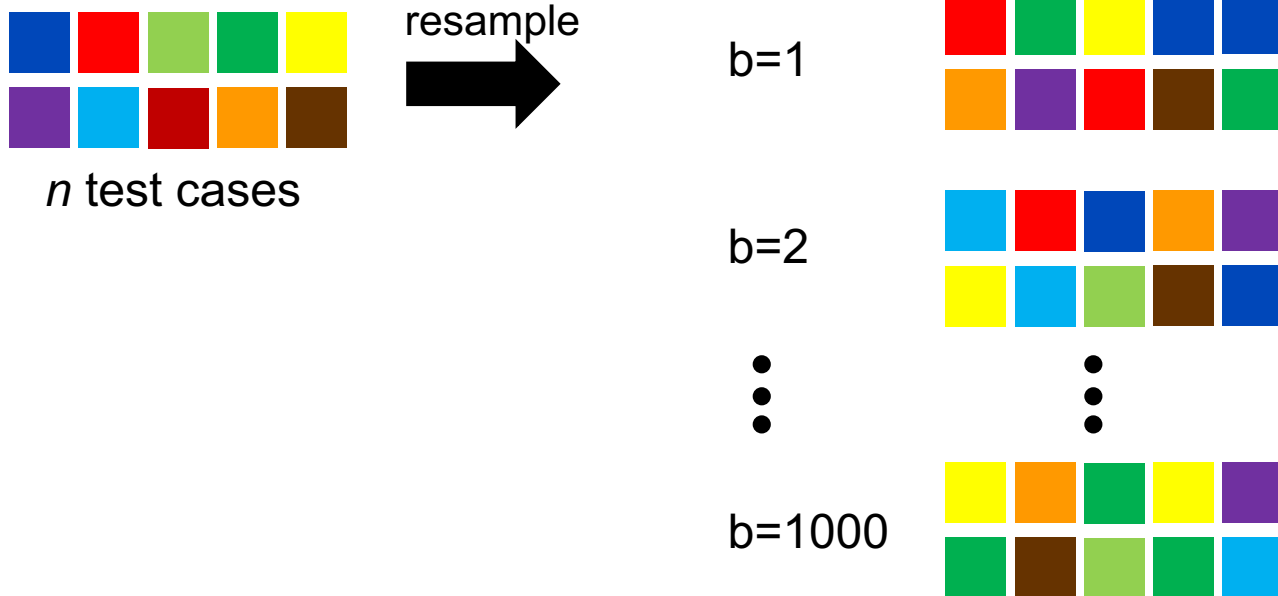
Significance map: Ranking uncertainty using hypothesis tests



- + Winner significantly superior to others?
- Difference between algorithms relevant?
- Little power to detect differences when # test cases small

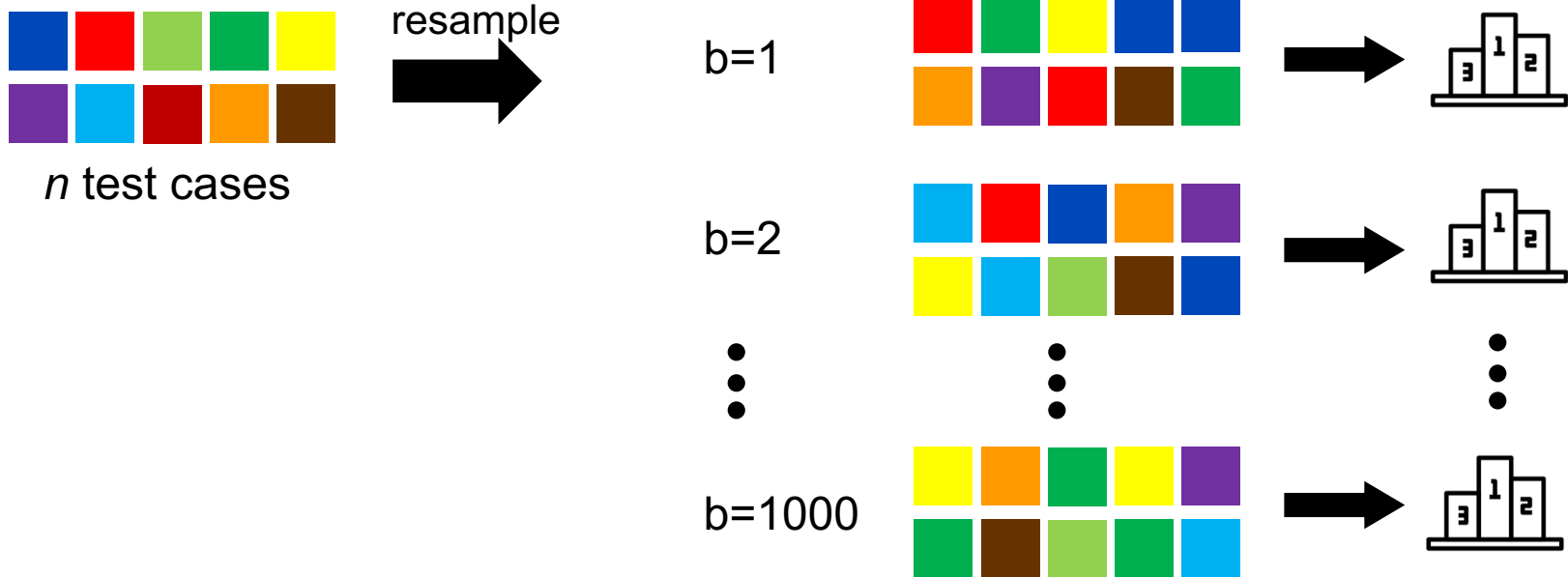
Ranking uncertainty using bootstrapping

1. Use available data set to generate 1000 bootstrap data sets



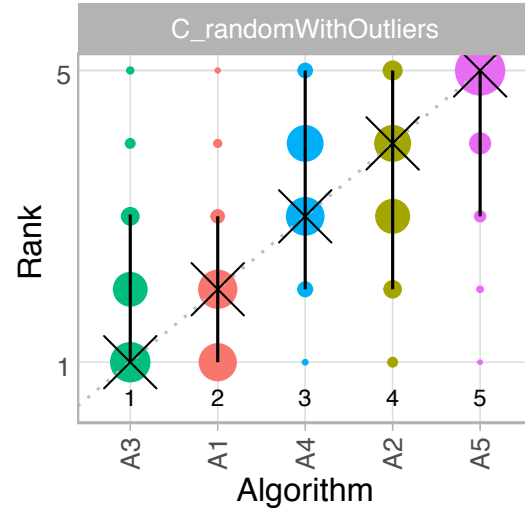
Ranking uncertainty using bootstrapping

1. Use available data set to generate 1000 bootstrap data sets
2. Perform ranking on each bootstrap data set



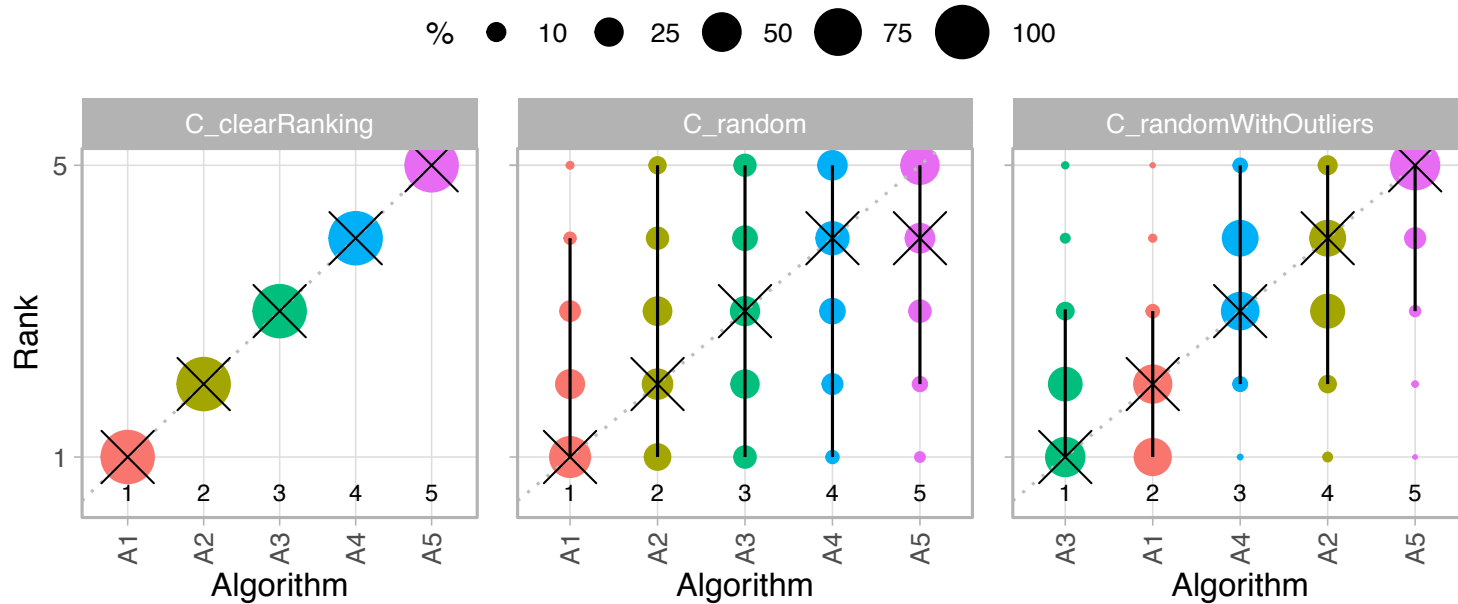
Blob plots: Ranking uncertainty using bootstrapping

% ● 10 ● 25 ● 50 ● 75 ● 100



+ What range of ranks for each algorithm is supported by the data?

Blob plots: Ranking uncertainty using bootstrapping



- + What range of ranks for each algorithm is supported by the data?
- Not sensible for very few test cases
- Difference between algorithms relevant?

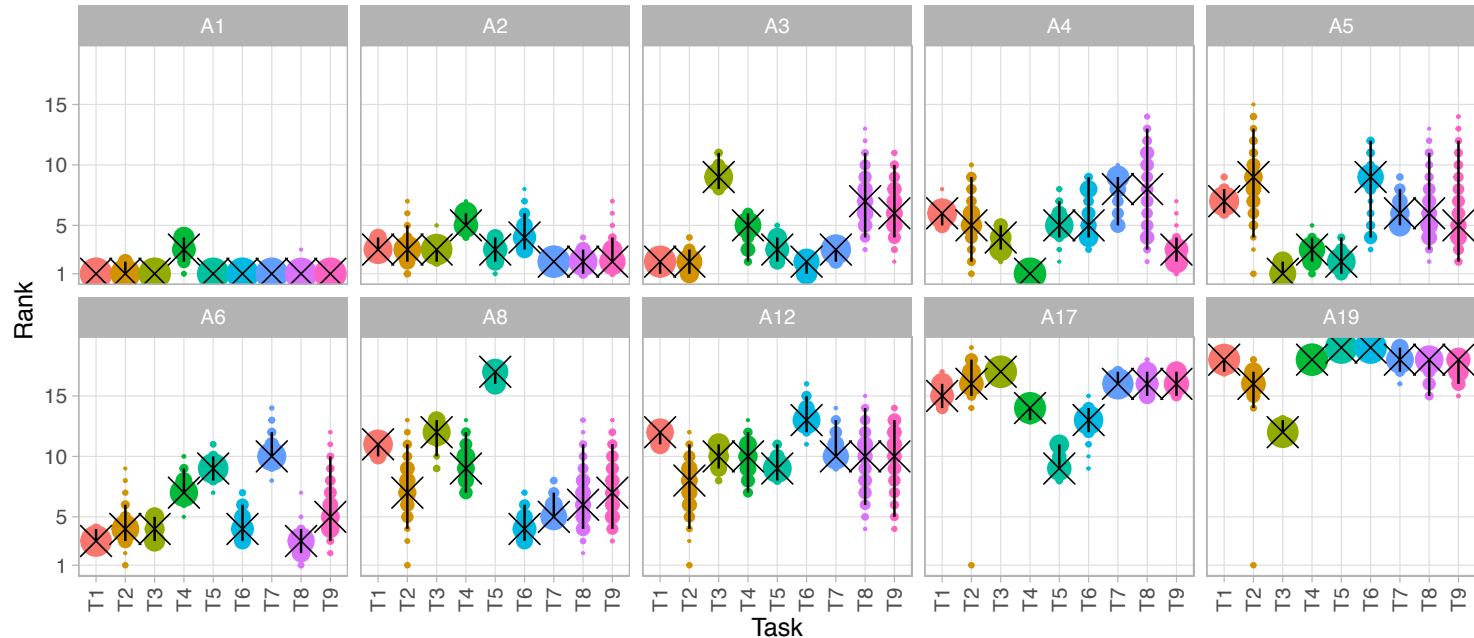
Multi-task challenges

Multi-task challenges

- Allow assessment of ...
 - generalizability of algorithms across tasks
 - which task yields clear separation of algorithms
 - similarity of tasks with regard to rankings
 - ...
- For illustration:
 - Subset of algorithms and tasks of *Medical Segmentation Decathlon* (2018)
 - Anonymized algorithms A_i , anonymized tasks T1-T9

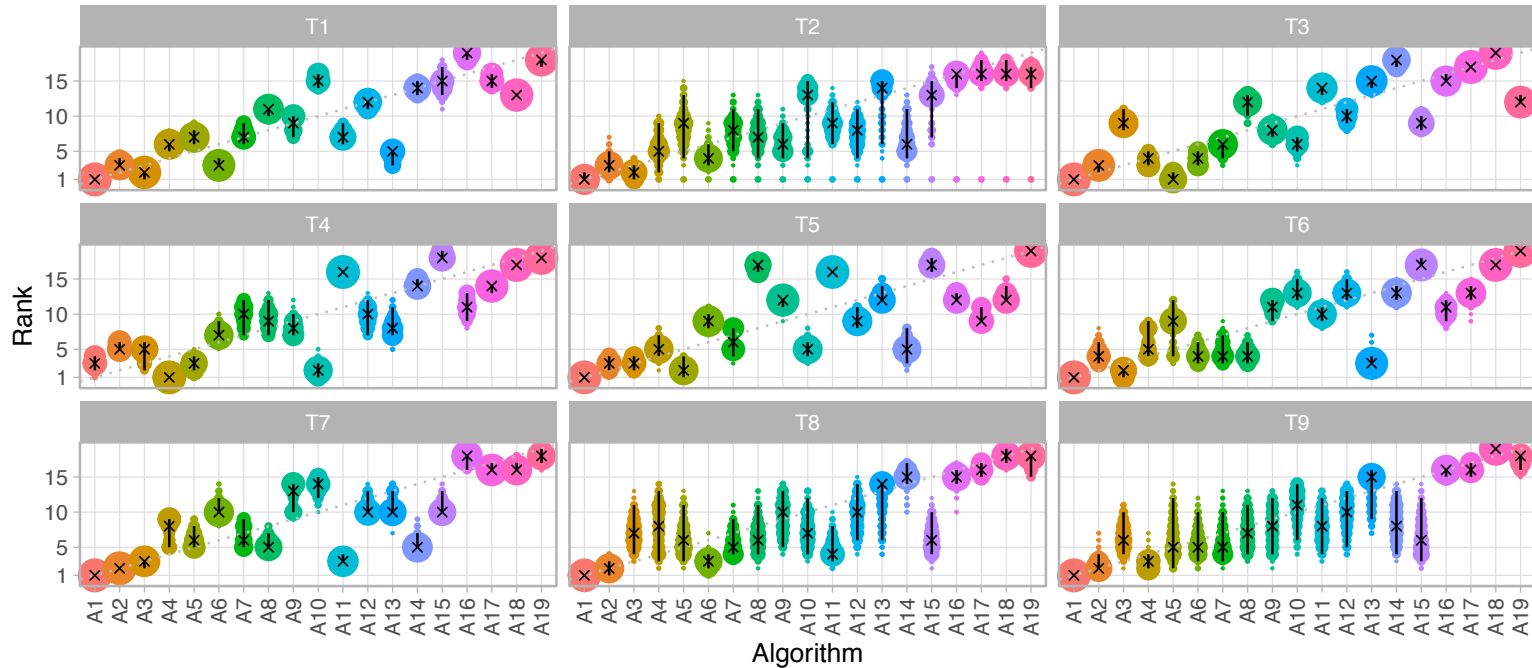
MSD 2018, <https://medicaldecathlon.com>

Blob plot stratified by algorithm



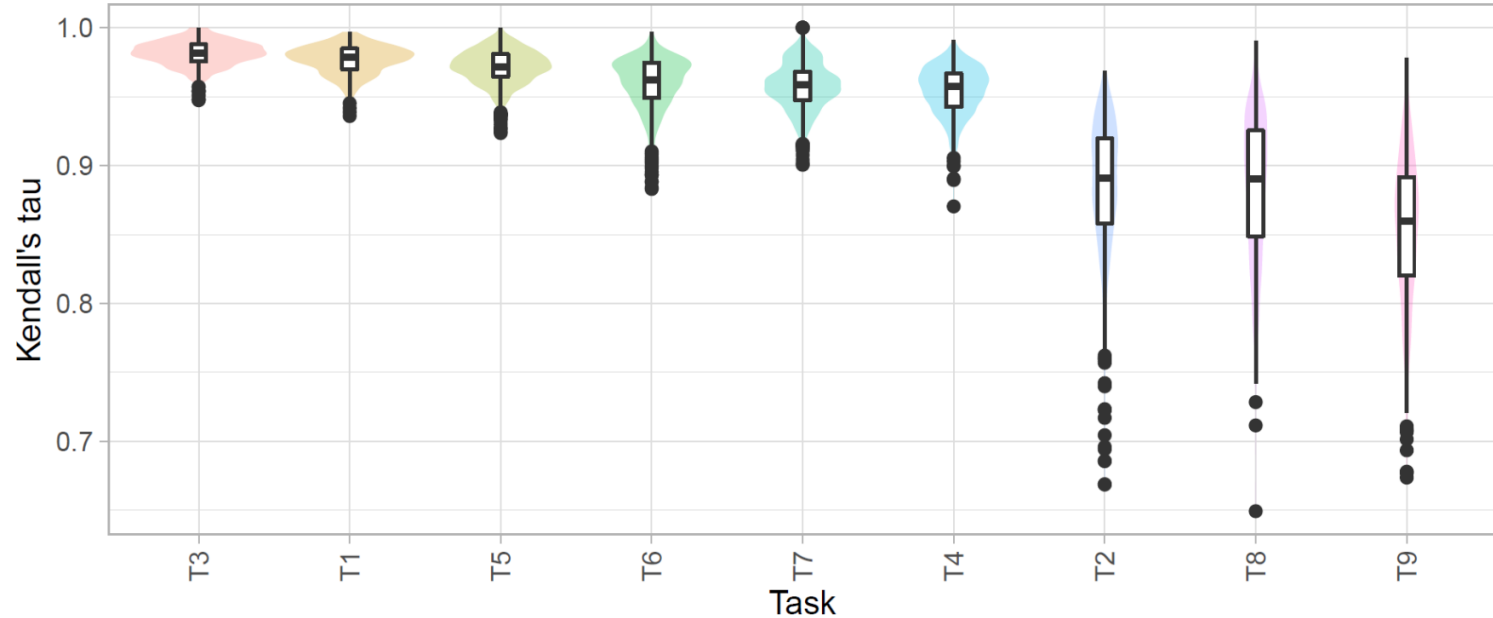
- + What range of ranks is supported by the data within and across tasks?
- + In which task does algorithm rank favorably or unfavorably?

Blob plot stratified by task



+ Which task yields clear separation of algorithms?

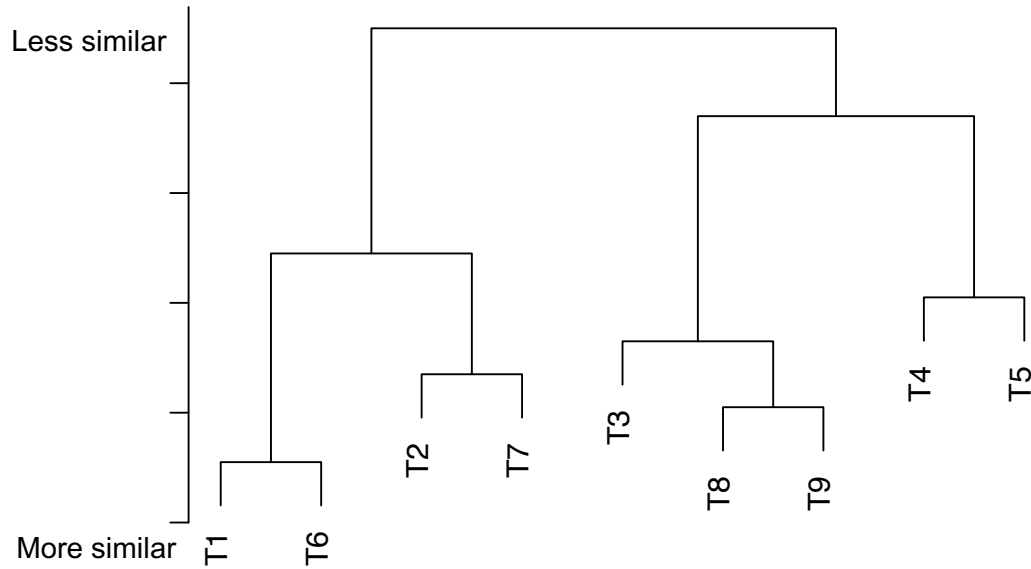
Violin plot: Overall ranking stability of a task



+ Compares ranking stability of entire ranking lists across tasks

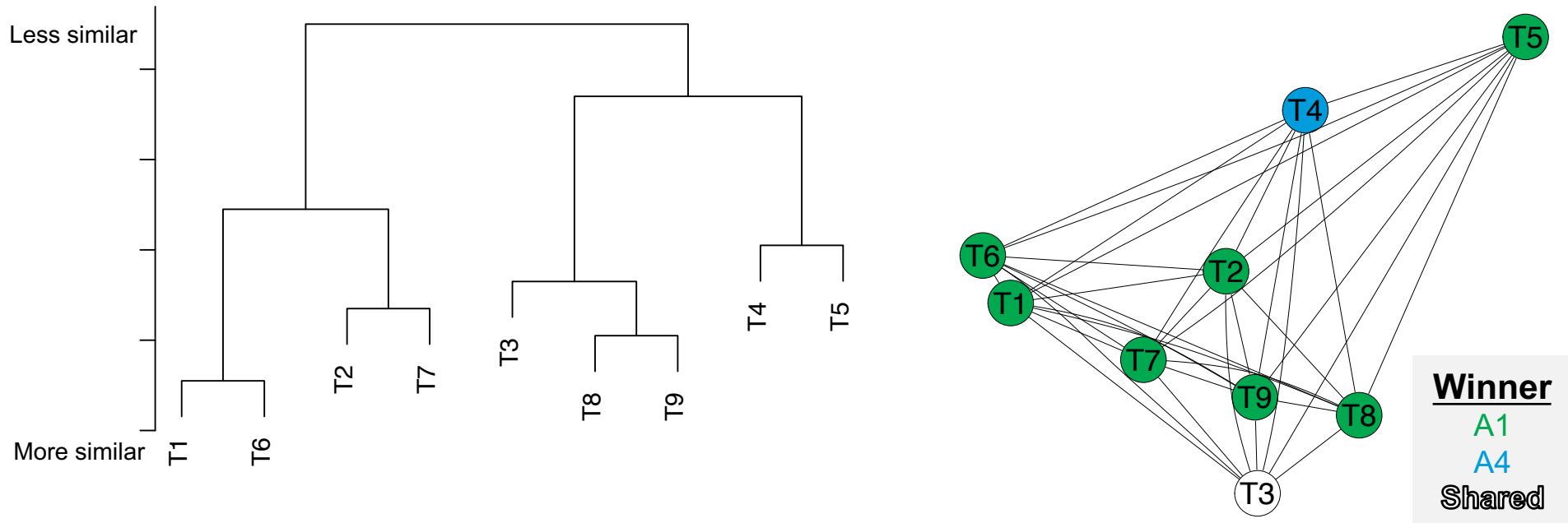
Dendrogram:

Similarity of tasks with regard to ranking



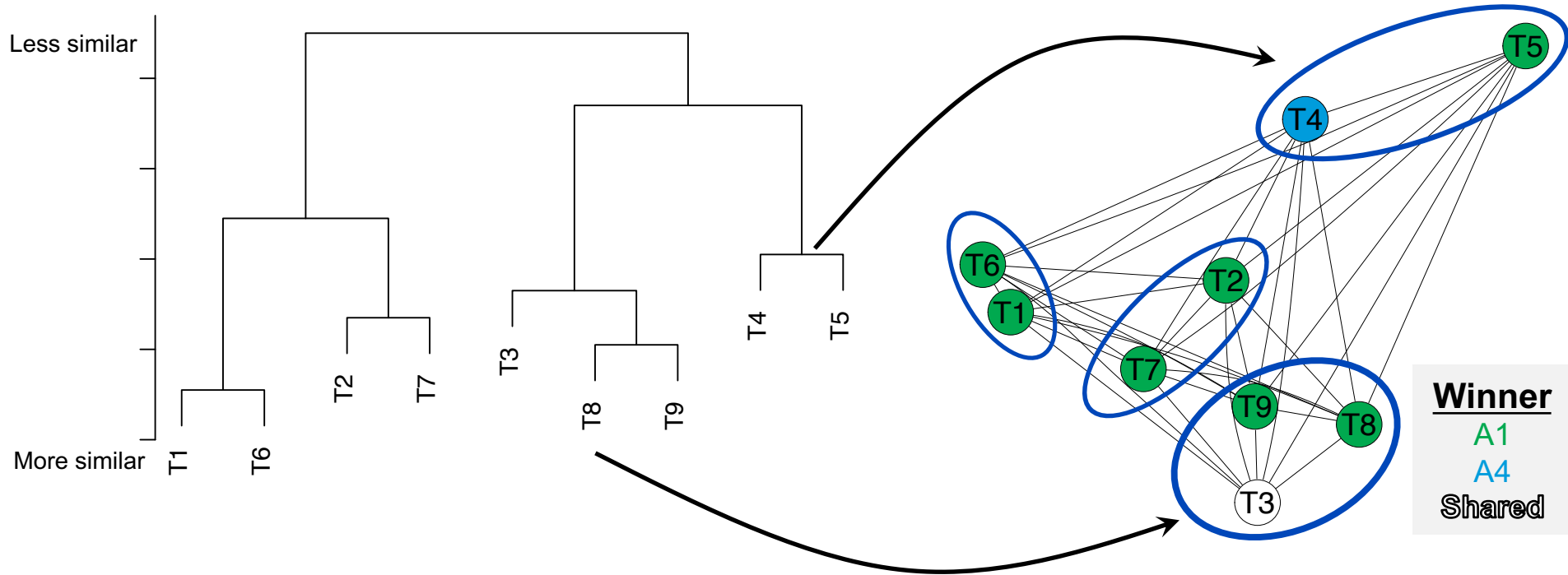
+ Group tasks according to similarity of their ranking lists

Dendrogram/Network: Similarity of tasks with regard to ranking



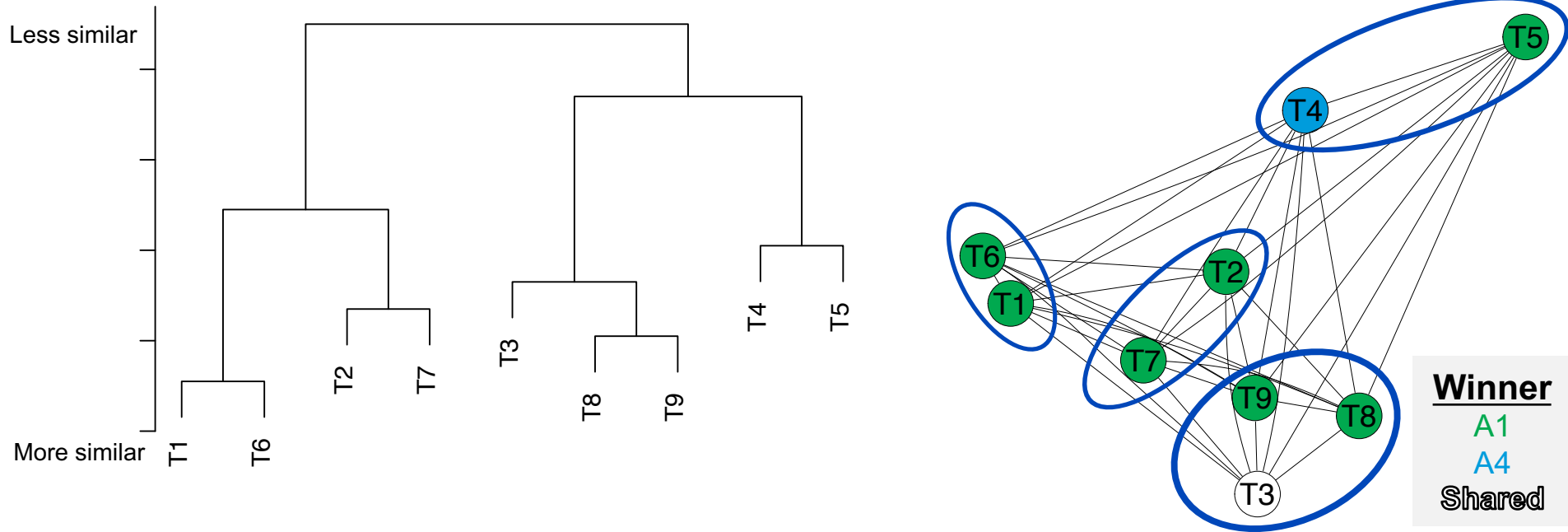
+ Group tasks according to similarity of their ranking lists

Dendrogram/Network: Similarity of tasks with regard to ranking



+ Group tasks according to similarity of their ranking lists

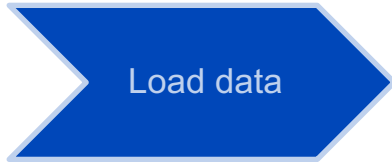
Dendrogram/Network: Similarity of tasks with regard to ranking



- + Group tasks according to similarity of their ranking lists
- Depend on chosen distance measure and agglomeration method/layout engine

Open-source framework: *challengeR*

Open-source toolkit *challengeR*



Testcase_ID	Algorithm_name	Metric_value	Task_name
85	A1	0.7952	c random
15	A4	0.6877	c clearRanking
81	A3	0.7754	c random
8	A5	0.6948	c random
82	A2	0.8576	c clearRanking
19	A2	0.5556	c random
84	A1	0.5215	c random

```
data_matrix = read.csv(file.choose())
challenge = data_matrix %>% as.challenge(
  case="Testcase_ID", algorithm="Algorithm_name",
  value="Metric_value",
  by="Task_name", # (only for multi-task)
  smallBetter = FALSE)
```

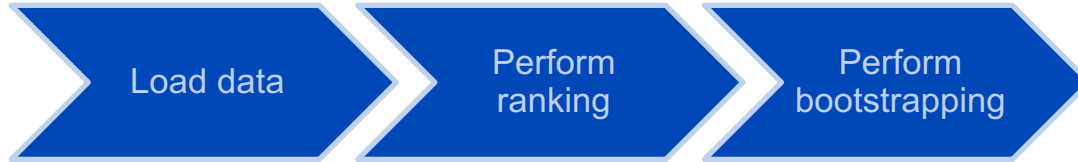
Open-source toolkit *challengeR*



```
ranking = challenge %>% testThenRank(...)
ranking = challenge %>% aggregateThenRank(...)
ranking = challenge %>% rankThenAggregate(...)

# Consensus ranking for multiple tasks:
meanRanks = ranking %>% consensus(method = "euclidean")
```

Open-source toolkit *challengeR*



```
ranking_bootstrapped = ranking %>% bootstrap(...)
```

Open-source toolkit *challengeR*



```
report(ranking_bootstrapped,  
       consensus=meanRanks,  
       format = "PDF",  
       ...)
```

Open-source toolkit *challengeR*



Alternatively all in a single call:

```
data_matrix %>% as.challenge(...) %>%  
  test(...) %>% rank(...) %>% bootstrap(...) %>% report(...)
```

Open-source toolkit *challengeR*



Alternatively all in a single call:

```
data_matrix %>% as.challenge(...) %>%  
  aggregate(...) %>% rank(...) %>% bootstrap(...) %>% report(...)
```

Open-source toolkit challengeR



Benchmarking report for Task1

created by challengeR 0.0.0.7 (Wiesenfarth, Runko, Carlson, Mairlein & Kopp-Schneider, 2019)

08 Oktober, 2019

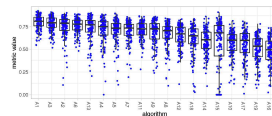
This document presents a systematic report in a benchmark study. Input data description, raw results values for all algorithms and test cases. Generated plots are:

- Visualization of assessment data: Heat map heatmap, violin plots and ranking heatmap
- Visualization of ranking robustness: Rank plots, Violin plots, significance maps and line plots
- Analysis based on test case: Heat map heatmap (1) ranking values

Algorithms are ordered according to chosen ranking scheme.

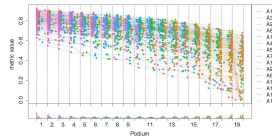
Ranking list:

alg	prop.	std.dev	rank
A1	0.042784	1	
A5	0.047208	2	
A2	0.050227	3	
A6	0.052127	4	
A13	0.054021	5	
A4	0.054166	6	
A5	0.054168	7	
A7	0.054168	8	
A11	0.054210	9	
A8	0.054342	10	
A10	0.054342	11	
A11	0.054342	12	
A9	0.054379	13	
A14	0.054390	14	
A12	0.054397	15	
A10	0.054400	16	
A12	0.054400	17	
A11	0.054401	18	
A14	0.054403	19	



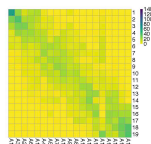
1.2 Violin plot

Violin plots (see also Engler et al. 2009) for visualizing raw assessment data. Upper part (gray/red) plot: Participating algorithms are color-coded, and each column lists the plot representative statistic value achieved with the respective algorithm. The actual statistic values are marked by the rank. Each plot (line, p=1) represents one possible rank, ordered from best (1) to last (19). The assignment of each value to the corresponding test case is visualized in the plot. The assignment of each value to the corresponding test case is visualized in the plot. Note that the plot part shows each column plot to further individualized 'Violins', where each column represents one participating algorithm (line p=10). This corresponding a vertical test case are connected by a line, leading to the chosen algorithm's direction. Lower part (purple) plot: The above represent the relative frequency for each algorithm to achieve the rank indicated by its position value.



1.3 Ranking heatmap

Ranking heatmap for visualizing raw assessment data. Each cell (i, A_j) shows the absolute frequency of test cases in which algorithm A_j achieved rank i .



2 Visualization of ranking stability

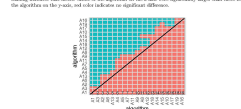
2.1 Heat plot for visualizing ranking stability

Algorithms are color-coded, and the area of each block at position $(A_i, rank_j)$ is proportional to the relative frequency A_i achieved each j across $n=1000$ bootstrap samples. The median rank for each algorithm is indicated by a black cross. 500 bootstrap intervals across bootstrap samples are indicated by black lines.



2.2 Violin plot for visualizing ranking stability based on bootstrapping

The results for boot cases for the 19 assessment data are presented. Each plot for ranking test based on the bootstrap ranking sample (line p=100 samples). The violin plot (ranked, bootstrap) contains: colored, Kennedy's τ is a solid index denoting the correlation between the test. It is computed by calculating the number of positive correlations and the difference between ranking lists and positive values between -1 (the lowest value) and 1 (the highest value). A violin plot, which continuously depicts a heatmap and a density plot, is generated from the results.



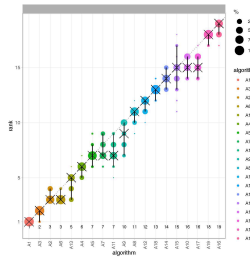
2.3 Significance maps for visualizing ranking stability based on statistical significance

Significance maps (heat matrices) visualize pairwise algorithm test results for the ranked Wilcoxon signed rank test as a 19x19 significance test with algorithms for each test case, according to the plot. Blue shading indicates that some values of the algorithm are statistically significantly higher than those from the algorithm in the same test and vice versa or no significant difference.



2.4 Ranking robustness with respect to ranking methods

Line plots for comparing ranking schemes across different ranking methods. Each algorithm is represented by one colored line. The rank testing result is marked on the x-axis. The height of the line represents the corresponding rank. Horizontal lines indicate shared ranks for all methods.



3 Reference

- Wiesenfarth, M., Runko, A., Carlson, M.J., Mairlein, L., and Kopp-Schneider, A. Analyzing and visualizing results of challengeR. *Journal of Statistical Software*.
- M. J. A. Engler, T. Böhner, and F. Lein, "Empirical and theoretical analysis of benchmark experiments," *Journal of Statistical Software*, International Statistical Institute, Germany, Technical Report 30, 2016. Online Available: [http://pubs.istat.it/archive, 4\(1\):1-11](http://pubs.istat.it/archive, 4(1):1-11).

Open-source toolkit *challengeR*: Single plots

```
ranking %>%
```

- `boxplot(...)`
- `podium(...)`
- `rankingHeatmap(...)`
- `significanceMap(...)`
- ...

```
ranking_bootstrapped %>%
```

- `violin(...)`
- ...

Summary

- Uncertainty and robustness of rankings needs to be assessed
- Open-source toolkit (R package)
 - Available on <https://github.com/wiesenfa/challengeR>
 - Computes rankings for diverse ranking methods
 - Generates an automatic report
- Wiesenfarth, Reinke, Landmann, Cardoso, Maier-Hein & Kopp-Schneider (2019). Methods and open-source toolkit for analyzing and visualizing challenge results. *arXiv preprint arXiv:1910.05121*

Summary

- Anyone already tried to use it?
- Feedback / problems / feature requests?
- Contributions welcome!

m.wiesenfarth@dkfz.de