Find similarity in Galaxy tools and predict next tools in workflows

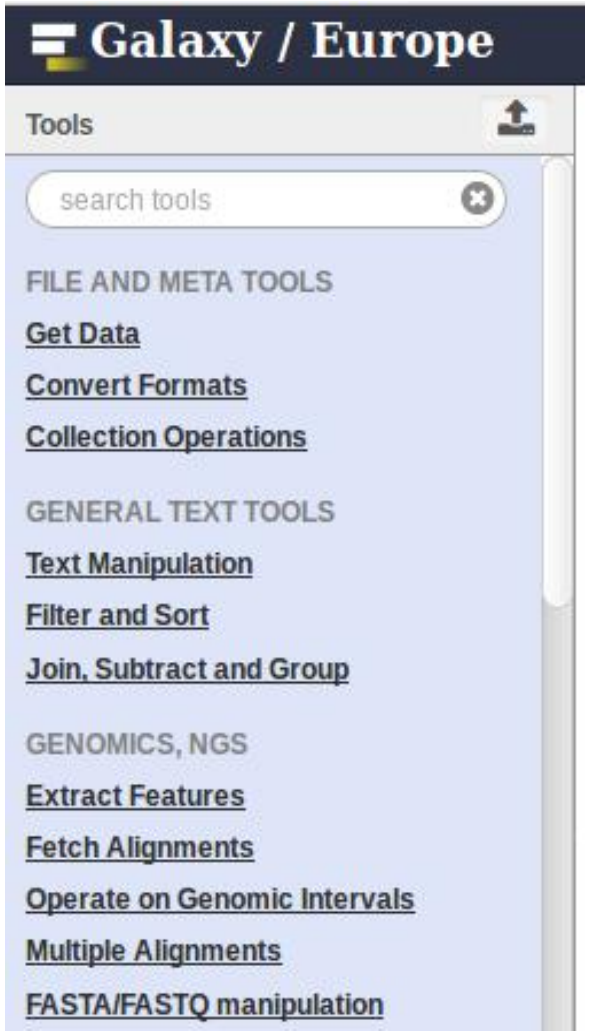
(Master's thesis)

Anup Kumar

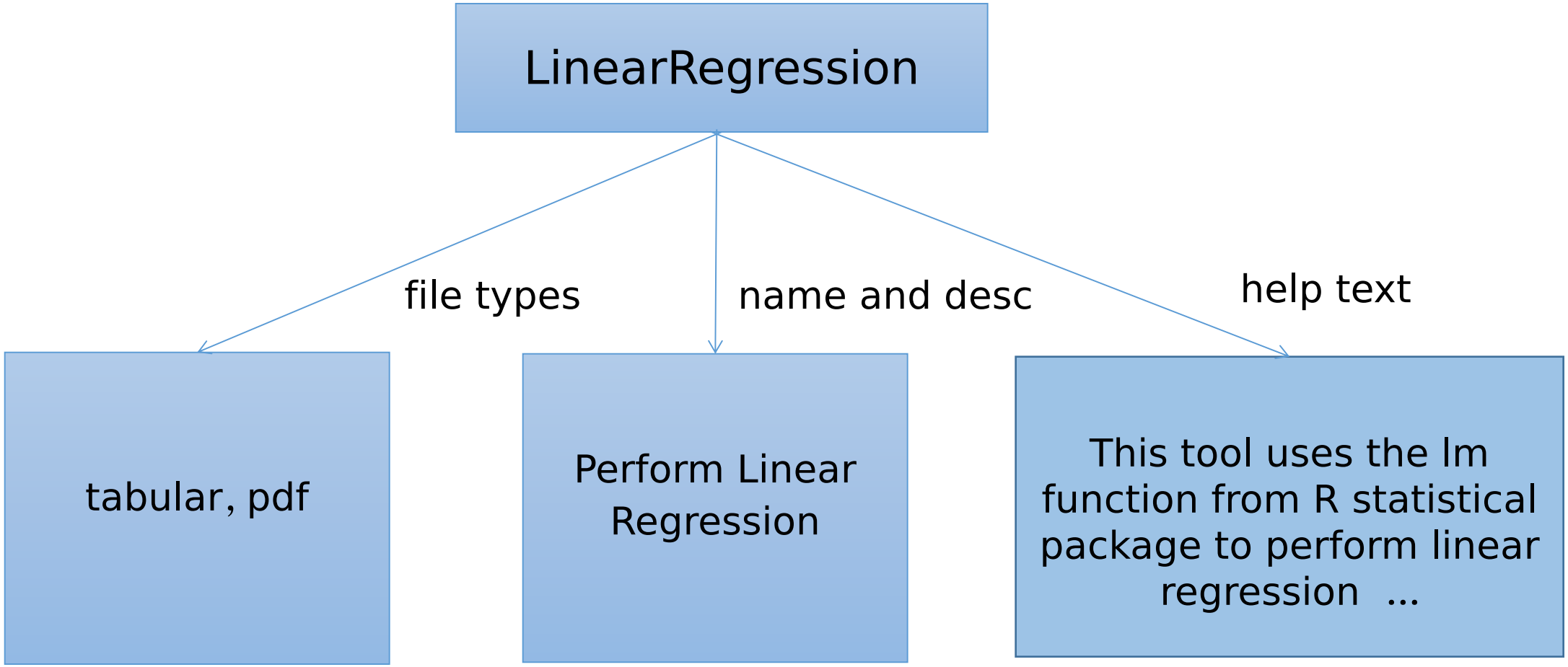
Find similarity in Galaxy tools

Machine learning (ML) and natural language processing (NLP) approaches

- Paragraph Vectors
- Gradient Descent



Tool’s attributes



Approach

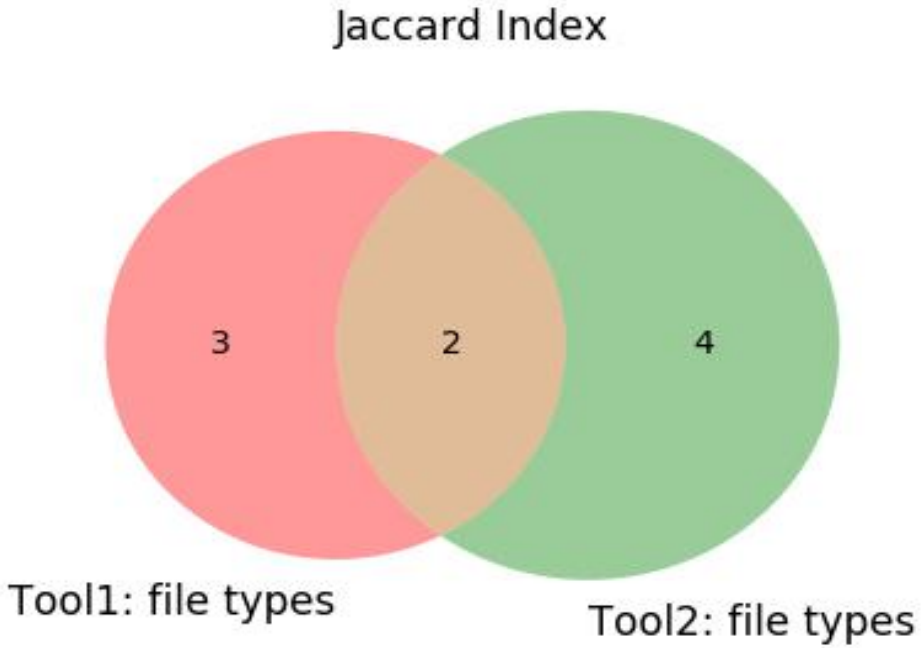
- Extract tool’s attributes
- Clean text
- Create sets of tokens
- Learn similarities
- Combine optimally
- Visualize

Tokens

Attributes/ Tools	LinearRegression	LogisticRegression	Similarity
Input, output	‘**pdf**‘, ‘**tabular**‘	‘**tabular**‘	?
Name, description	‘**regress**‘, ‘**linear**‘, ‘**perform**‘	‘**logist**‘, ‘**regress**‘, ‘**perform**‘	?
Help text	‘**regress**‘, ‘**assumpt**‘, ‘**lm**‘, ‘**statist**‘,‘**linear**‘ …	‘**vif**‘,‘**regress**‘, ‘**glm**‘,‘**car**‘,‘**inflat**‘, ‘**function**‘,‘**statist**‘, ‘**logist**‘ …	?

Compute similarity

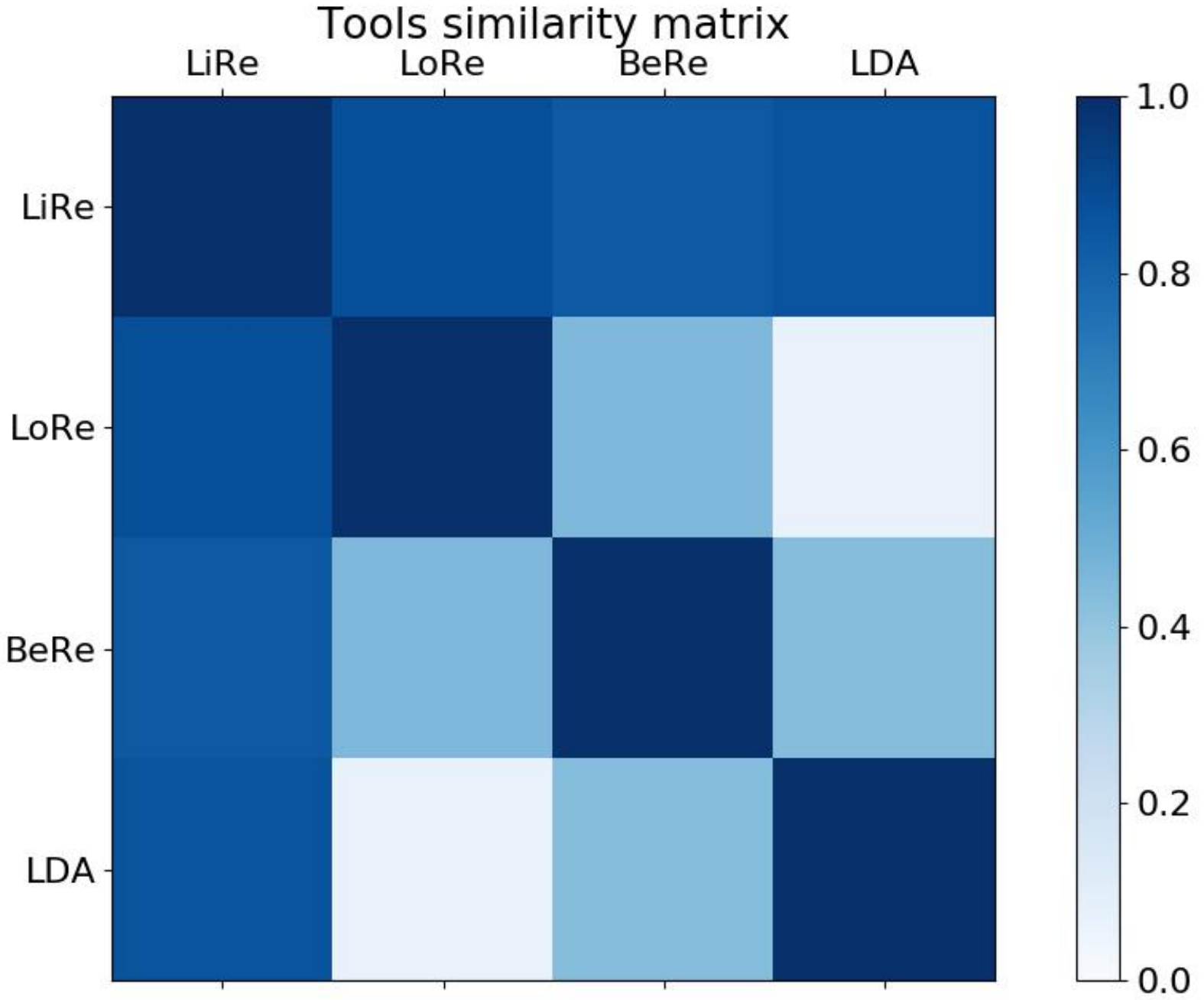
- Compute Jaccard Index for input, output



- Learn dense vectors for name, desc. and help text*

  [‘regress‘, ‘linear‘, ‘perform‘] = [$0.98, 0.07, \ldots, 0.12$]
- Compute cosine distance between dense vectors

*[https://cs.stanford.edu/~quocle/paragraph_vector.pdf]



- LiRe - Linear Regression
- LoRe - Logistic Regression
- BeRe - BestSubsetsRegression
- LDA - LDA Analysis

How to combine？

- 3 similarity matrices, one for each attribute
- How to combine them？Take average？
- Optimal combination, learn weights for each tool
- Similarity:

$$\underset{(w_i,\ldots,w_n)}{\arg\max} \sum_{i=1}^{N} w_i \cdot s_i$$

Optimization

s1 Input, output

1	0.34	0.65	0.44
0.34	1	...	...
0.65	...	1	...
0.44	...	...	1

s2 Name, desc.

1	0.76	0.63	0.85
0.76	1	...	...
0.63	...	1	...
0.85	...	...	1

s3 Help text

1	0.06	0.1	0.17
0.06	1	...	...
0.1	...	1	...
0.17	...	...	1

$$Minimize\left(\left[1.0\,,\,1.0\,,\,1.0\,,\,\ldots\,,\,1.0\right]\,-\,\left[w1\cdot s1\,+\,w2\cdot s2\,+\,w3\cdot s3\right]\right)$$

where $w1 + w2 + w3 = 1$

Example

- Tool: LinearRegression
- Similarity for input/output: **s1** = [$1.0, 0.34, 0.65, 0.44$]
- Similarity for name, desc: **s2** = [$1.0, 0.76, 0.63, 0.85$]
- Similarity for help text: **s3** = [$1.0, 0.06, 0.1, 0.17$]
- Optimal weights: **w1** = 0.3, **w2** = 0.6, **w3** = 0.1
- Similarity:

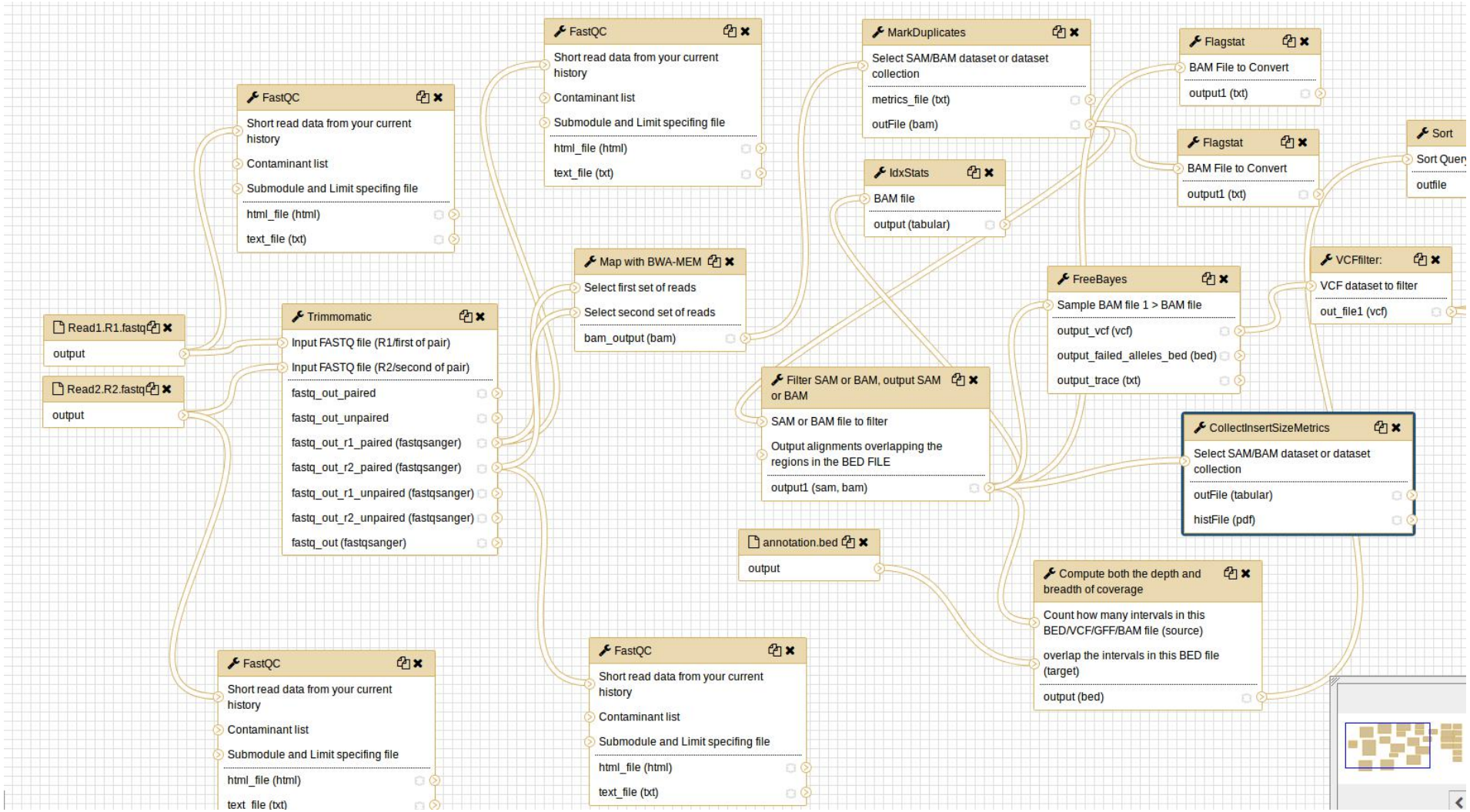
$$[w1 \cdot s1 + w2 \cdot s2 + w3 \cdot s3]$$

Visualizer and References

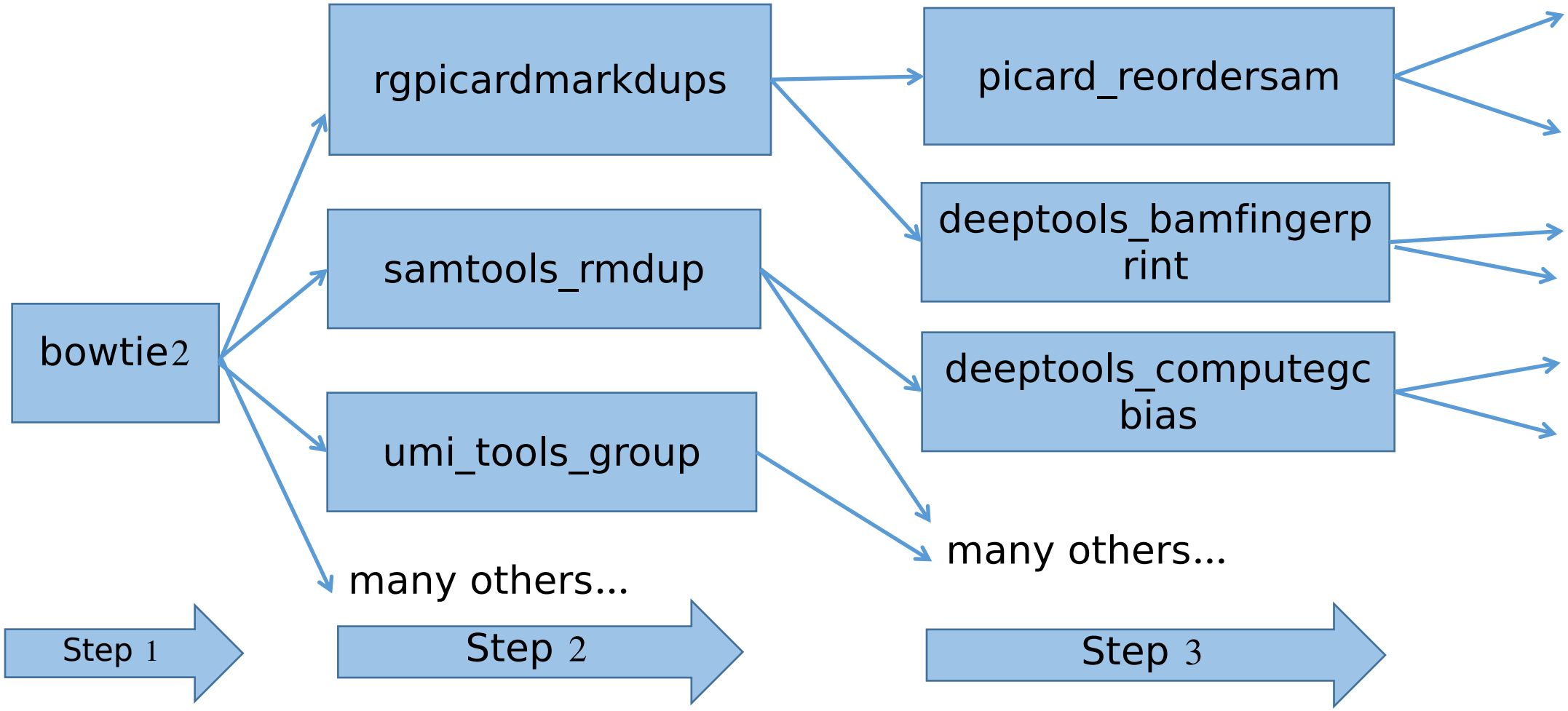
- Static website: results for ~ 1000 tools
- https://rawgit.com/anuprulez/similar_galaxy_tools/master/viz/similarity_viz.html
- https://github.com/anuprulez/similar_galaxy_tools
- https://cs.stanford.edu/%7Equocle/paragraph_vector.pdf
- https://arxiv.org/pdf/1607.05368.pdf

Predict next tools in Galaxy workflows

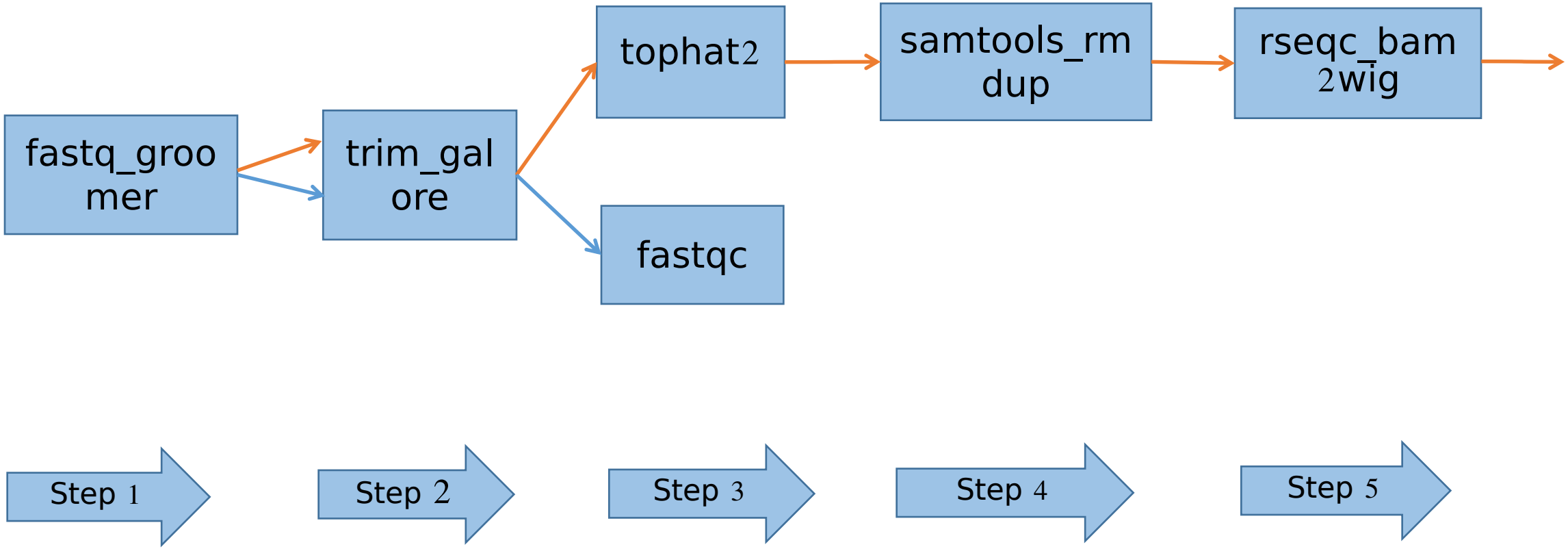
Galaxy workflow



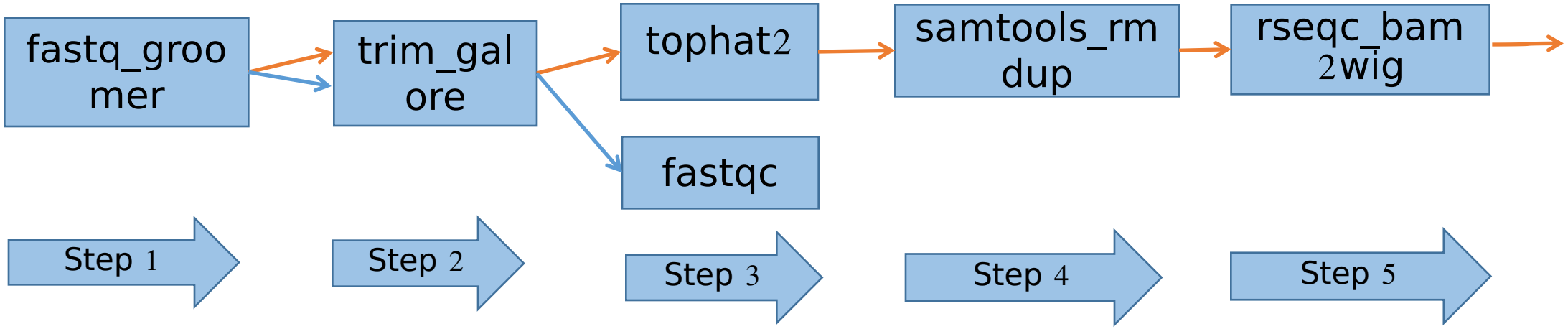
Next tools ?


rgpicardmarkdups
samtools_rmdup
umi_tools_group
many others...
bowtie2
picard_reordersam
deeptools_bamfingerprint
deeptools_computegcbias
many others...
Step 1
Step 2
Step 3

Workflow as a sequence



Data preprocessing

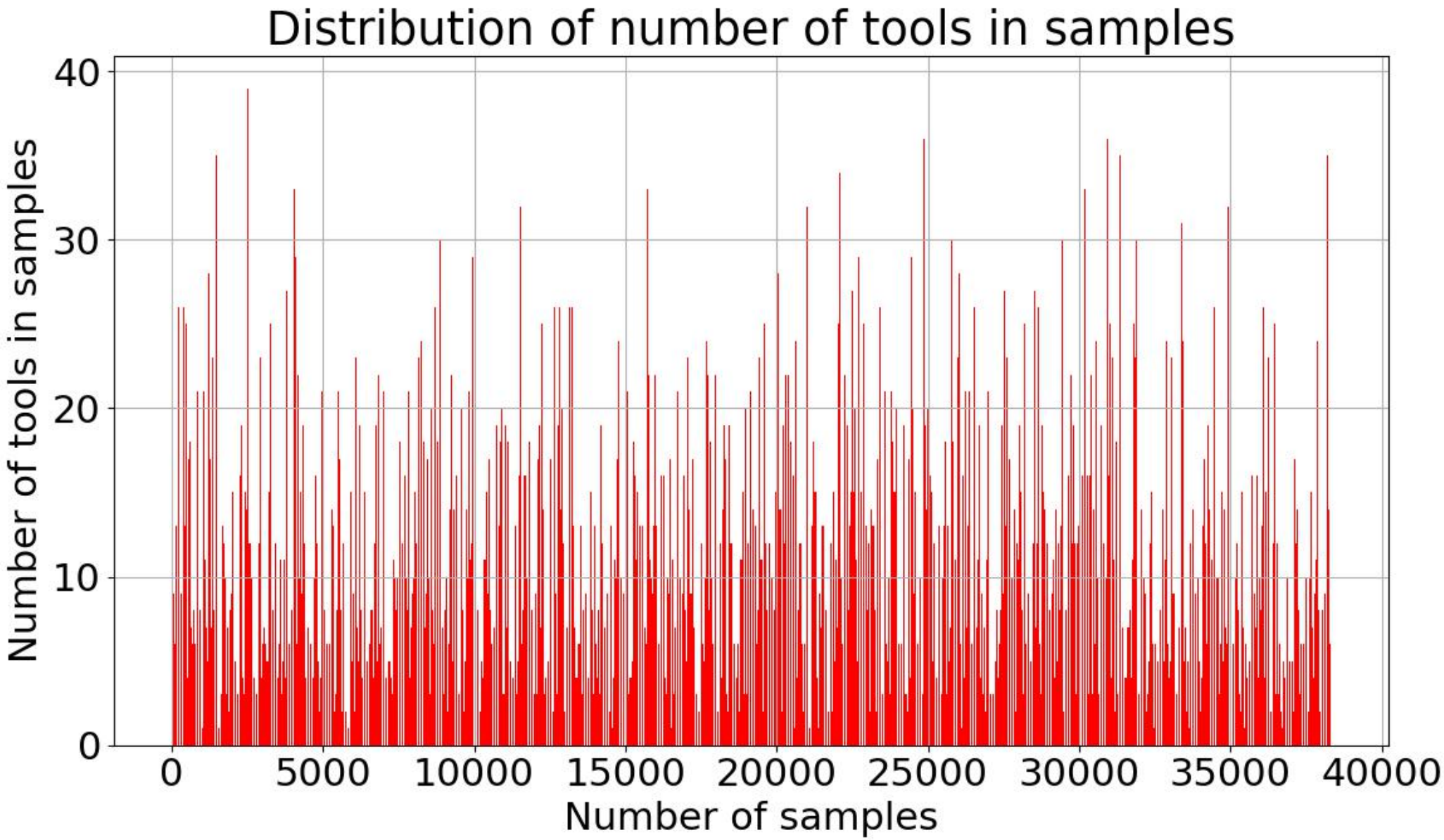


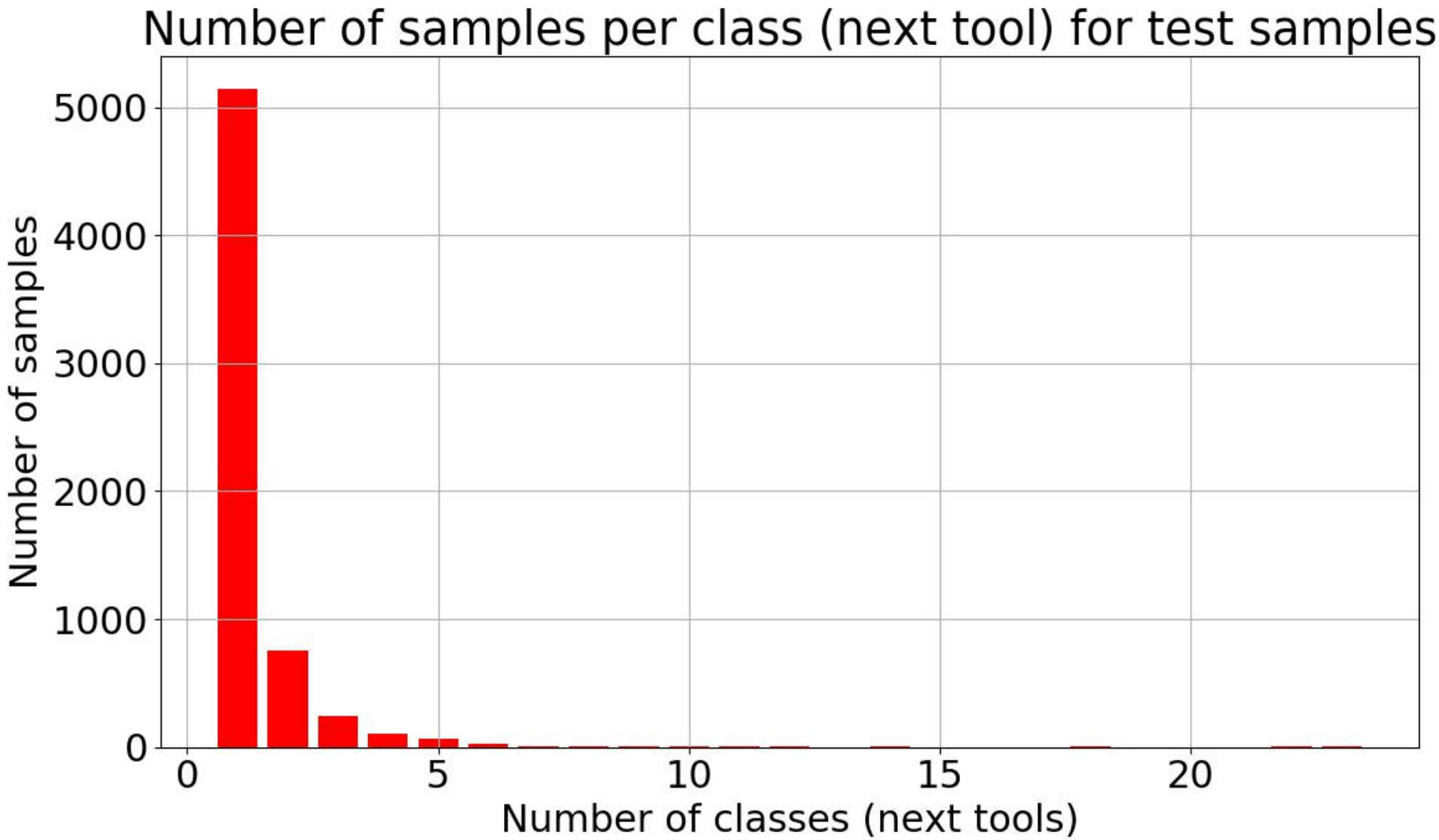
- fastq_groomer, trim_galore (Step 1)
- fastq_groomer, trim_galore, tophat2, fastqc (Step 2)
- fastq_groomer, trim_galore, tophat2, samtools_rmdup (Step 3)
- fastq_groomer, trim_galore, tophat2, samtools_rmdup, rseqc_bam2wig (Step 4)

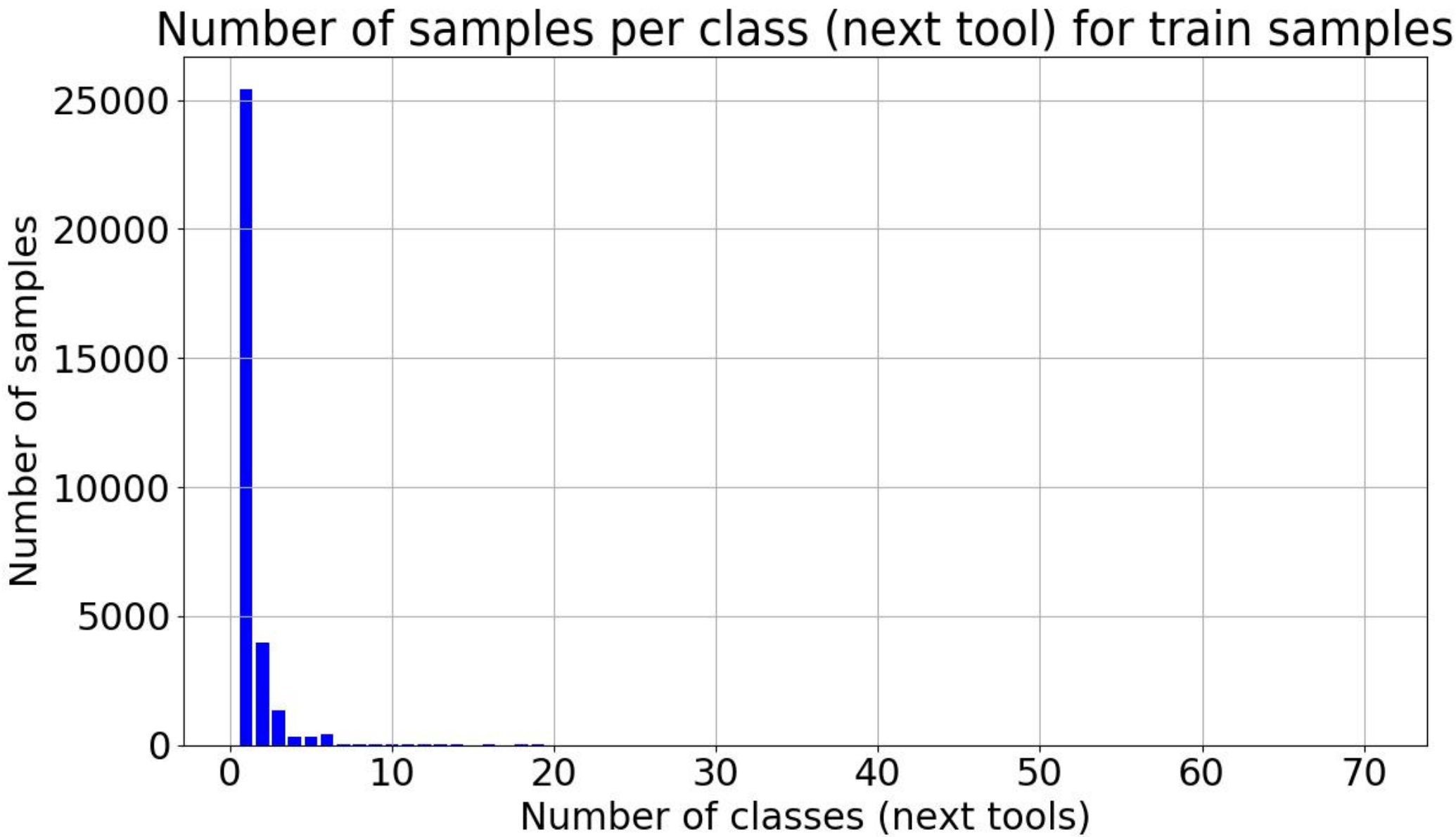
Data preprocessing

‘fastq_groomer‘: 1, ‘trim_galore‘: 2, ‘tophat2‘: 3,
‘samtools_rmdup‘: 4, ‘rseqc_bam2wig‘: 5, ‘fastqc‘: 6

Sample	Label (next tool(s)/classes)
1,2 (fastq_groomer, trim_galore)	3, 6 (tophat2, fastqc)
1,2,3	4
….	….
….	….

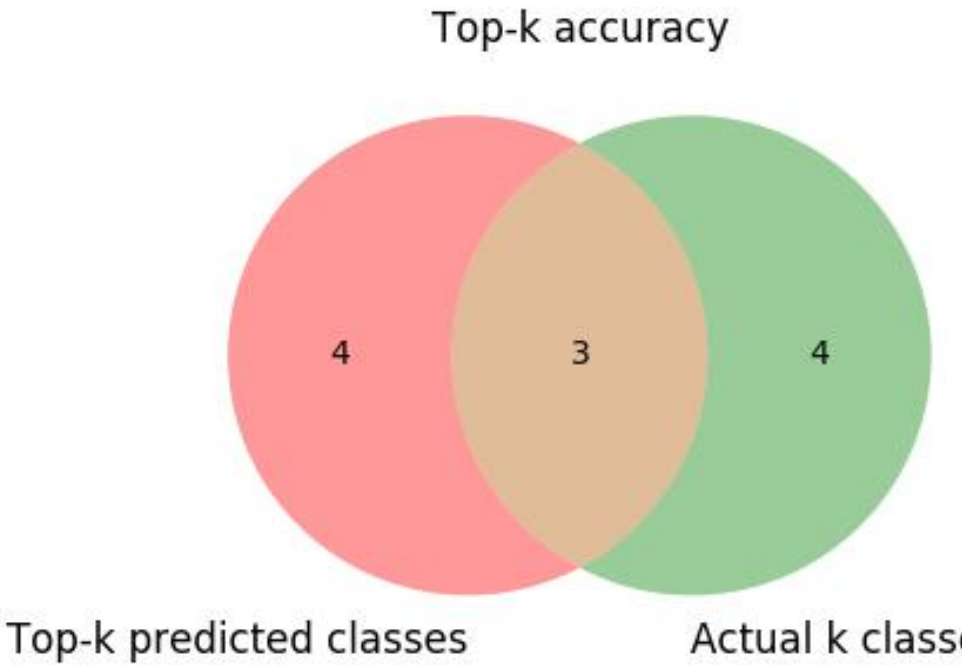

Distribution of number of tools in samples
Number of tools in samples
0
10
20
30
40
0
5000
10000
15000
20000
25000
30000
35000
40000
Number of samples

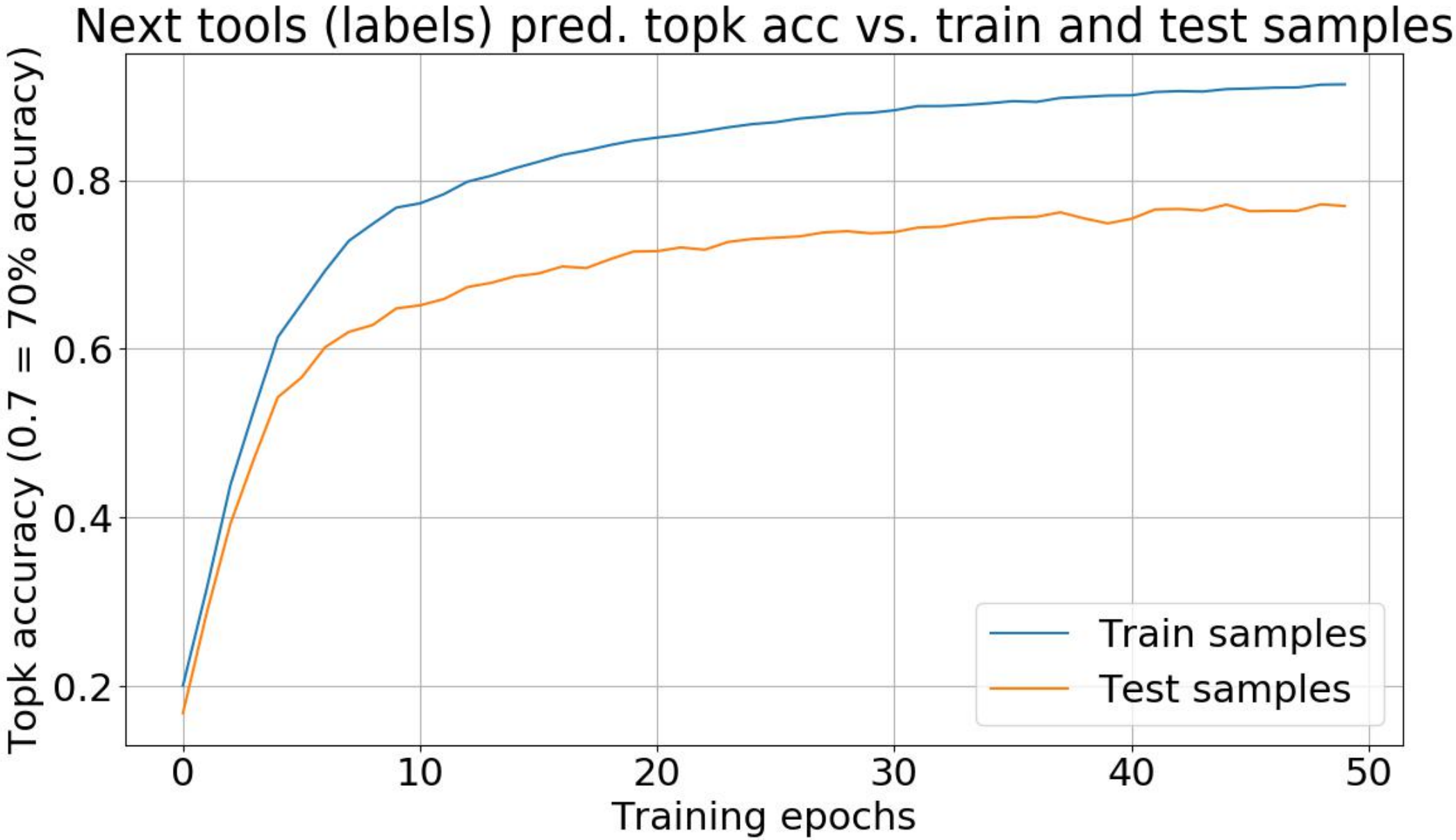

Number of samples per class (next tool) for test samples
Number of samples
5000
4000
3000
2000
1000
0
0
5
10
15
20
Number of classes (next tools)

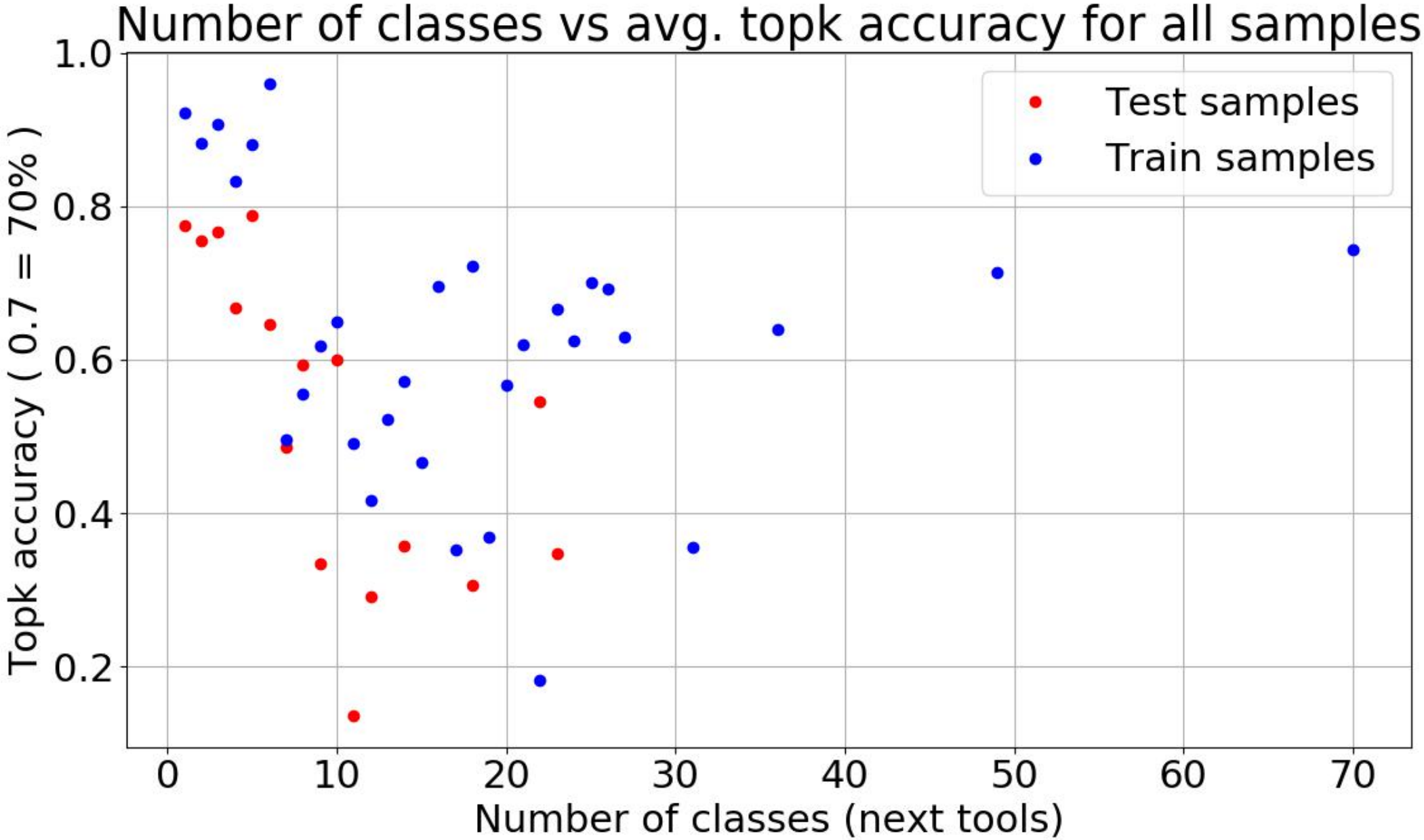

Number of samples per class (next tool) for train samples
Number of samples
25000
20000
15000
10000
5000
0
0
10
20
30
40
50
60
70
Number of classes (next tools)

Classification

- Multi label, multi class classification
- Long short term memory (LSTM) networks
- Topk accuracy




Next tools (labels) pred. topk acc vs. train and test samples
Topk accuracy (0.7 = 70% accuracy)
0.2
0.4
0.6
0.8
0
10
20
30
40
50
Training epochs
Train samples
Test samples


Number of classes vs avg. topk accuracy for all samples
Topk accuracy (0.7 = 70%)
Number of classes (next tools)
Test samples
Train samples
1.0
0.8
0.6
0.4
0.2
0
10
20
30
40
50
60
70

References

- https://github.com/anuprulez/similar_galaxy_workflow
- https://arxiv.org/pdf/1511.03677.pdf
- https://arxiv.org/pdf/1604.04573.pdf
- https://arxiv.org/pdf/1506.00019.pdf

Thank you for your attention

Questions？