

Experimental Validation

The Motor Imagery (MI) **BCI FII corpus** is used to validate the Julia implementation of the Riemannian Minimum Distance to mean (MDM) classifier in *PosDefManifold.jl* against the reference implementations in Python (*PyRiemann*). The classification accuracy and the execution time in the two implementations is compared.

Two MI classification tasks are considered:

- 1) “right hand” vs. “feet”,
- 2) “left hand” vs. “right hand”.

Classification Accuracy

The pre-processing and encoding have been carried-out separately in the two languages. For each session in all databases, an 8-fold cross-validation has been performed using the *StratifiedKFold* function of the Python *Scikit-learn* package to extract the training and validation splits, which have been used identically in both languages. This ensures that exactly the same splits are used in the two languages.

Pre-processing

Pass-band 4th-order linear-phase response (forward-backward) Butterworth digital filter in the region 8Hz-32Hz.

Encoding

Sample covariance matrix estimation of the MI trials with a Tikhonov regularization (diagonal loading) with parameter 10e-4.

Model fitting and prediction

Riemannian MDM (nearest centroid).

Table 1: Resources for the implementation in Julia and Python

Implementation	Julia	Python
Pre-processing	DSP.jl (from Eegle.jl)	SciPy
Encoding	Eegle.jl	PyRiemann
Model fitting and prediction	PosDefManifoldML.jl	PyRiemann

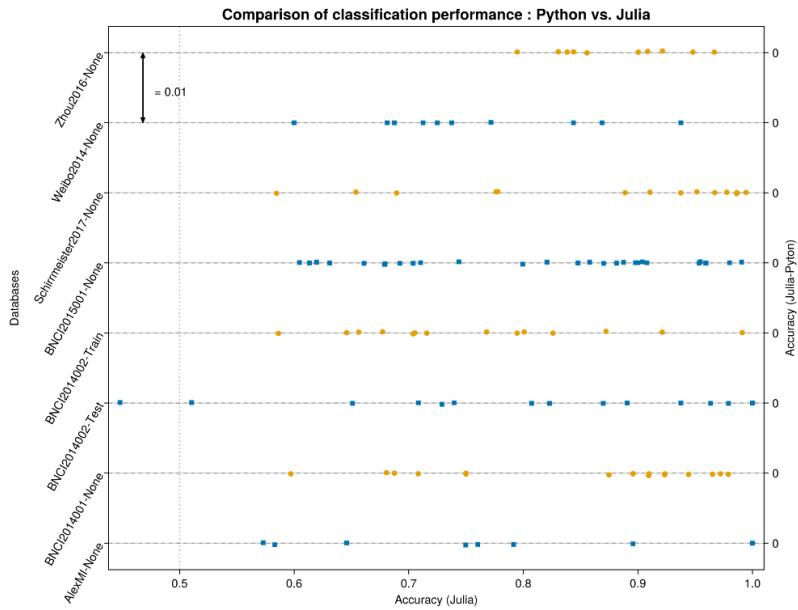
Results

1) “right hand” vs “feet”

Table 2: Classification accuracy for “right hand” vs. “feet” MI

Database	Julia Mean and Std (%)	Python Mean and Std (%)	Difference of Means
AlexMI-None	74.99 ± 14.90871	75.0 ± 14.9092	-0.009375
BNCI2014001-None	84.94 ± 11.90432	84.95 ± 11.9096	-0.011944
BNCI2014002-Test	78.98 ± 17.0275	78.98 ± 17.03021	-0.001179
BNCI2014002-Train	76.18 ± 11.3729	76.17 ± 11.36971	0.005857
BNCI2015001-None	81.11 ± 12.84454	81.1 ± 12.8419	0.001214
Schirrmeister2017-None	86.3 ± 13.98145	86.29 ± 13.98154	0.005071
Weibo2014-None	75.66 ± 10.0781	75.65 ± 10.07801	0.0005
Zhou2016-None	87.85 ± 5.3869	87.84 ± 5.38551	0.009591

Figure 1: Classification accuracies for “right hand” vs. “feet” MI. Each point is a session. Points lying on the horizontal dashed lines indicate a null difference of the accuracy obtained in the two languages.

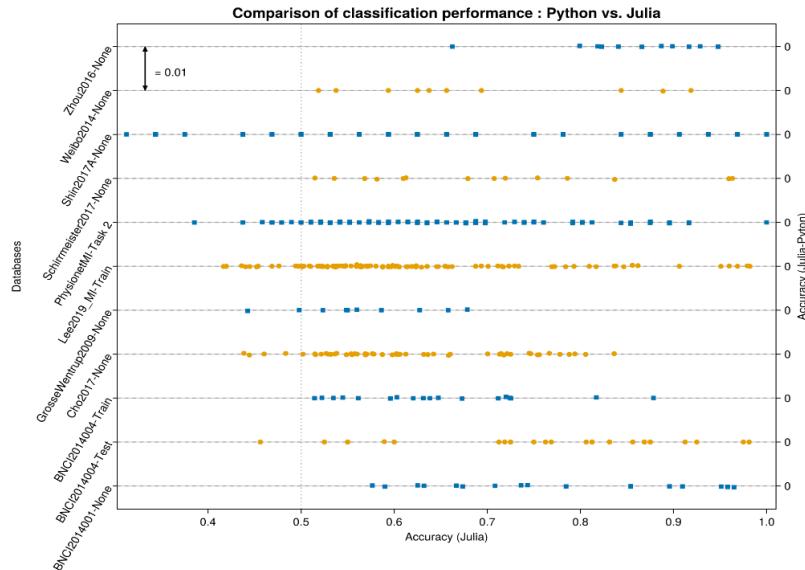


2) “left hand” vs. “right hand”.

Table 3: Classification accuracy for left hand vs. right hand MI

Database	Julia Mean and Std (%)	Python Mean and Std (%)	Difference
BNCI2014001-None	78.24 ± 13.80576	78.24 ± 13.81513	-0.004611
BNCI2014004-Test	76.53 ± 14.12921	76.53 ± 14.12962	0.000315
BNCI2014004-Train	64.8 ± 10.11533	64.8 ± 10.11401	0.002833
Cho2017-None	62.1 ± 10.29322	62.1 ± 10.29349	-3.8e-5
GrosseWentrup2009-None	56.72 ± 7.26641	56.72 ± 7.26111	0.00045
Lee2019_MI-Train	62.0 ± 13.2963	62.0 ± 13.29557	-0.001285
PhysionetMI-Task 2	62.09 ± 12.82857	62.09 ± 12.83081	-0.000216
Schirrmeister2017-None	70.2 ± 14.54348	70.2 ± 14.54554	-0.002179
Shin2017A-None	63.25 ± 16.43238	63.25 ± 16.43238	0.0
Weibo2014-None	69.14 ± 14.35384	69.14 ± 14.35598	-0.0014
Zhou2016-None	85.35 ± 7.97634	85.35 ± 7.97701	0.001682

Figure 2: Classification accuracies for left hand vs. right hand MI. Each point is a session. Points lying on the horizontal dashed lines indicate a null difference of the accuracy obtained in the two languages.



Execution Time

The execution time for performing the above cross-validations has been estimated as the median execution time observed across five repetitions of the whole cross-validation procedure, excluding pre-processing and encoding. Both in the Julia and Python implementation, multithreading has been enabled, allowing four threads to be used in parallel.

Results

1) “right hand” vs “feet”

Table 4: Execution time in milliseconds for “right hand” vs. “feet” MI

Database	Julia Mean and Std (ms)	Python Mean and Std (ms)	Difference (abs)
AlexMI-None	14 ± 2	256 ± 3	242
BNCI2014001-None	114 ± 45	437 ± 29	323
BNCI2014002-Test	26 ± 27	258 ± 11	232
BNCI2014002-Train	35 ± 11	288 ± 26	253
BNCI2015001-None	55 ± 23	285 ± 31	230
Schirrmeister2017-None	13454 ± 16005	76923 ± 54557	63469
Weibo2014-None	1971 ± 1965	4802 ± 2655	2956
Zhou2016-None	38 ± 32	283 ± 17	245

Figure 3: Execution Time in milliseconds (\log_{10}) for “right hand” vs. “feet” MI

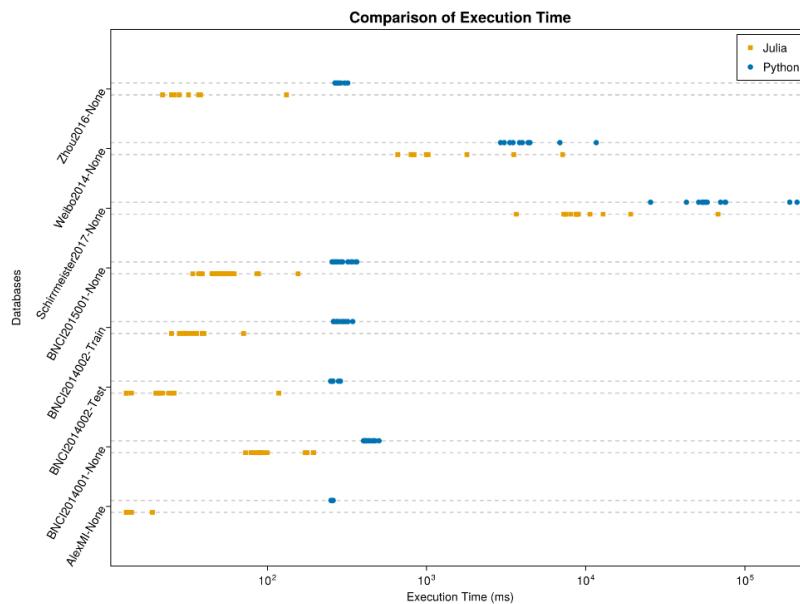
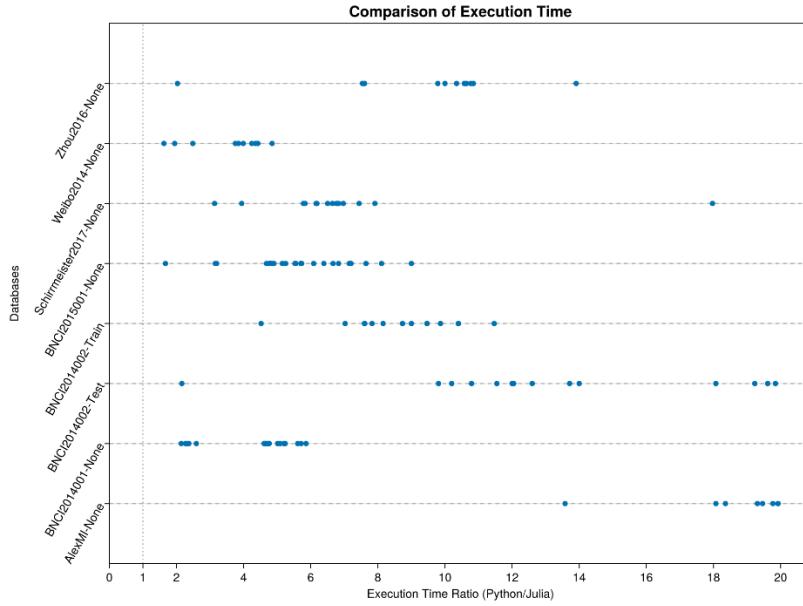


Figure 4: Execution Time (Ratio Python/Julia) for “right hand” vs. “feet” MI



2) “left hand” vs. “right hand”.

Table 5: Execution time in milliseconds for “left hand” vs. “right hand” MI

Database	Julia Mean and Std (ms)	Python Mean and Std (ms)	Difference (abs)
BNCI2014001-None	109 ± 37	406 ± 11	296
BNCI2014004-Test	15 ± 4	260 ± 4	245
BNCI2014004-Train	12 ± 3	264 ± 8	251
Cho2017-None	943 ± 62	4049 ± 269	3106
GrosseWentrup2009-None	3940 ± 375	30792 ± 3249	26852
Lee2019_MI-Train	599 ± 104	2552 ± 526	1953
PhysionetMI-Task	333 ± 270	931 ± 204	856
Schirrmeister2017-None	12647 ± 14830	76416 ± 54510	63769
Shin2017A-None	25 ± 8	277 ± 11	252
Weibo2014-None	1823 ± 1953	4665 ± 2624	2842
Zhou2016-None	37 ± 8	273 ± 4	236

Figure 5: Execution Time in milliseconds (\log_{10}) for “left hand” vs. “right hand” MI

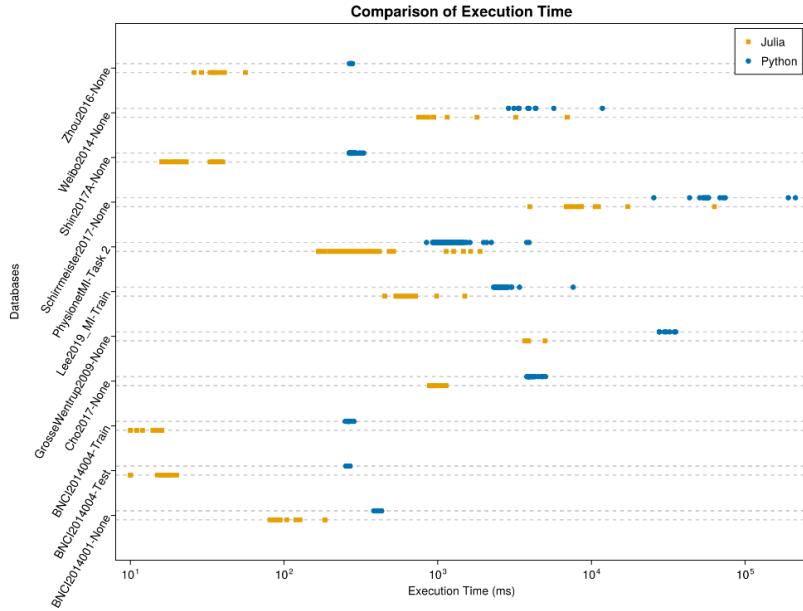
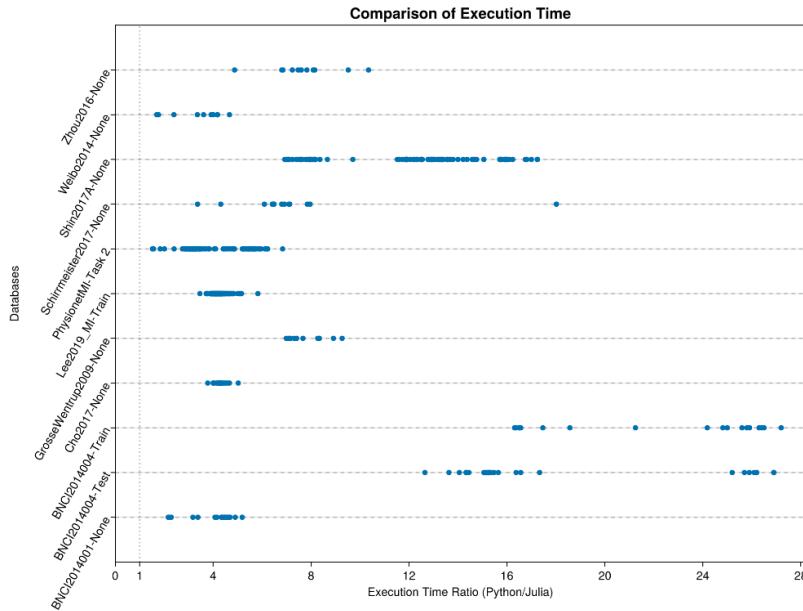


Figure 6: Execution Time (Ratio Python/Julia) for left hand vs. right hand MI



Conclusions

The Julia and Python implementations allow the same accuracy. Typically, the Julia implementation is a few times faster. The speed-up factor with the analyzed data is up to ≈ 27 .