Game-Benchmark for Evolutionary Algorithms

Vanessa Volz*, Boris Naujoks+, Tea Tušar′, Pascal Kerschke#

*TU Dortmund University, Germany
+TH Köln - University of Applied Sciences, Germany
′Jožef Stefan Institute, Slovenia
#WWU Münster University, Germany

15th July 2018

Game Benchmark: But Why?

- On the one hand:
  Multiple game-related competitions at GECCO and CIG for algorithms, no systematic analysis and comparison.

- On the other hand:
  Benchmarking analysis tools based on artificial testfunctions. Now: Game-Benchmark!

OK... and HOW?

- Part 1: Problems
  1. Collect game-related problems
  2. Integrate them with COCO
  3. Analyse results
  4. Make the benchmark available publicly

- Part 2: Discussions
  1. Organise a workshop
  2. Discuss the benchmark with **YOU**

Cool! WHAT can I do?

- Request problem characteristics
  `https://ls11-www.cs.tu-dortmund.de/people/volz/gamesbench_part.html#char`
- Contribute your game-related problem
  Open an issue `https://github.com/ttusar/coco`
- Run your algorithm on the benchmark
  Get the code `https://github.com/ttusar/coco`
- Join in our discussion

Table of Contents

A Short Introduction to COCO

Tea Tušar

Computational Intelligence Group
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia

July 15, 2018

Workshop on Game-Benchmark for Evolutionary Algorithms
Genetic and Evolutionary Computation Conference, GECCO 2018
Kyoto, Japan

Why benchmark optimization algorithms?

No free lunch theorem $\Rightarrow$ No algorithm works best for all optimization problems

Purpose of benchmarking: To be able to select the best algorithm for the given real-world optimization problem

Preconditions

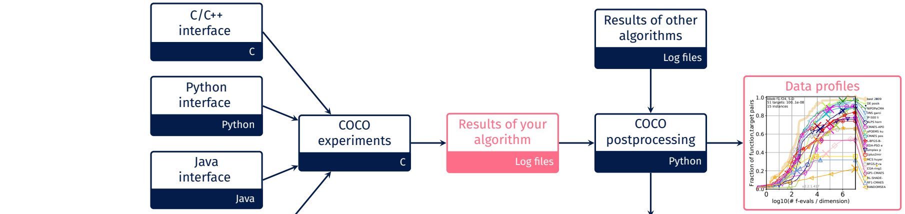
- The real-world problem with some known properties
- Test problems with similar properties to those of the real-world problem
- Results of several optimization algorithms on these test problems for any number of evaluations

How to benchmark optimization algorithms?

The COCO platform

- COCO (Comparing Continuous Optimizers)
- **`https://github.com/numbbo/coco`**
- Automatized benchmarking of optimization algorithms
  - Test problems with known properties
  - Data of previously run algorithms available for comparison
  - Provides interfaces to C/C++, Python, Java, Matlab/Octave
- Being developed at Inria Saclay, France, since 2007

Benchmarking with COCO



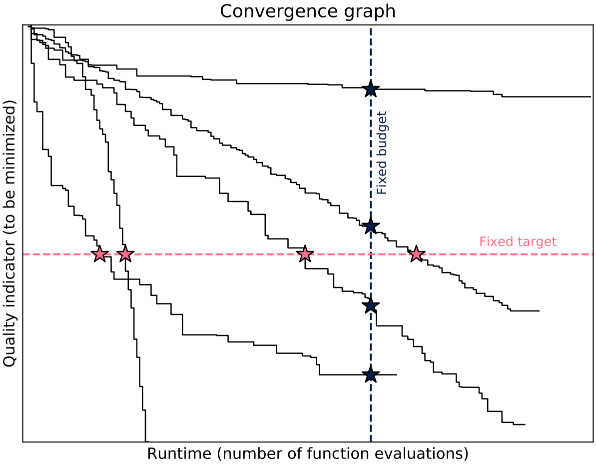
Tables

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	12	15/15
ALPS ho	9.2(13)	140(35)	304(45)	499(63)	693(58)	1050(38)	1436(82)	15/15
EDA-PSO	3.2(2)	20(9)	324(184)	901(166)	1524(146)	2733(386)	3899(487)	15/15
MCS huy	**1**(0.1)	**1.9**(2)	**2.5**(2)	**2.6**(2)	**2.6**(2)	**2.6**(2)	**2.6**(2)	15/15
VNS gar	7.4(3)	19(3)	26(4)	32(5)	39(3)	51(4)	65(5)	15/15
1plus2m	**1.5**(1.0)	5.0(1)	8.1(2)	12(1)	15(3)	22(3)	28(3)	15/15
oPOEMS	102(45)	139(15)	334(147)	779(201)	1215(164)	2034(352)	2843(178)	15/15
CMAES p	**2.3**(1)	8.7(3)	15(6)	22(5)	28(4)	41(4)	53(6)	15/15
DE posi	5.0(4)	21(9)	39(9)	61(11)	83(7)	124(9)	167(15)	15/15
NIPOPaC	**2.7**(2)	7.5(3)	14(2)	20(4)	26(3)	37(3)	51(5)	15/15
CGA-rin	6.7(7)	28(10)	67(13)	154(46)	271(88)	2085(1031)	4.7e4(4e4)	0/15
IP-500	3.0(1.0)	7.6(1)	13(3)	21(2)	27(3)	41(7)	54(6)	15/15
simplex	21(0.7)	64(28)	77(30)	84(18)	87(15)	92(18)	96(12)	15/15
L-BFGS-	**1.4**(0.0)	**1.3**(0.0)	**1.3**(0.0)	**1.4**(0.3)	**1.4**(0.3)	**1.4**(0.2)	**1.4**(0.3)	15/15
RL-SHAD	**2.4**(3)	9.3(4)	14(5)	20(3)	26(7)	41(24)	205(323)	1/15
GP1-CMA	**2.3**(2)	6.1(1)	9.2(2)	13(3)	16(4)	21(5)	31(5)	15/15
RF1-CMA	**2.6**(1)	8.3(2)	15(2)	25(4)	44(16)	217(245)	758(1022)	1/15
RANDOMS	3.2(4)	1037(4718)	1.9e5(9e4)	2.7e7(4e7)	∞	∞	∞ *5e7*	0/15
CMAES-A	**2.3**(3)	6.0(3)	14(3)	19(3)	26(1)	39(5)	52(4)	15/15
BFGS-P-	**1.5**(0)	**1.3**(0)	**1.3**(0)	**1.3**(0)	**1.3**(0)	**1.3**(0)	**1.8**(0.6)	15/15

Requirements: C compiler and Python (other languages are optional)

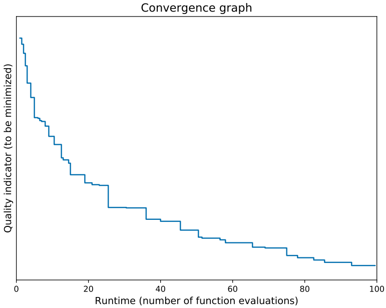
The fixed-target approach

Interested in the runtime (number of function evaluations) needed to achieve a target value



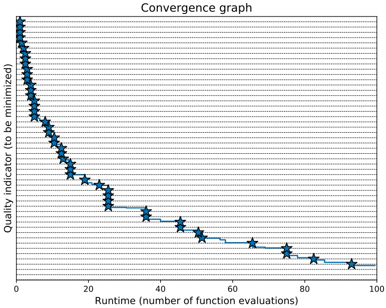
Data profile

The data profile is the empirical cumulative distribution function (ECDF) of the recorded runtimes



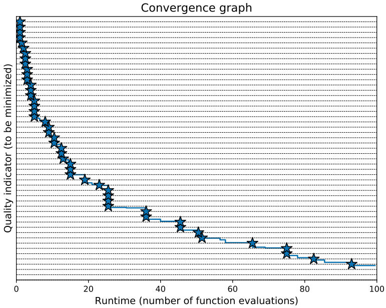
Data profile

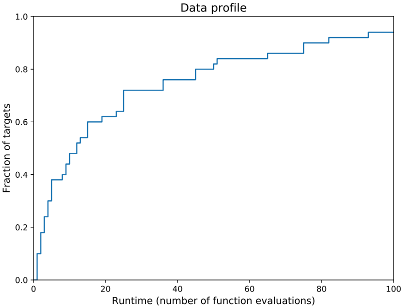
The data profile is the empirical cumulative distribution function (ECDF) of the recorded runtimes



Data profile

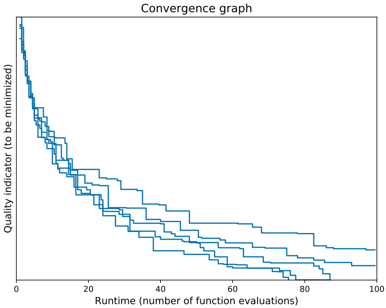
The data profile is the empirical cumulative distribution function (ECDF) of the recorded runtimes





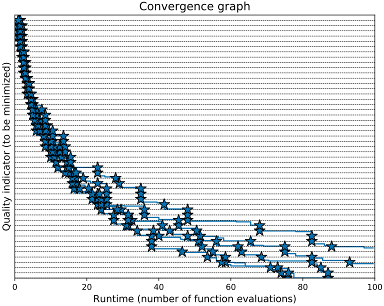
Data profile

Data profiles can aggregate performance over multiple runs



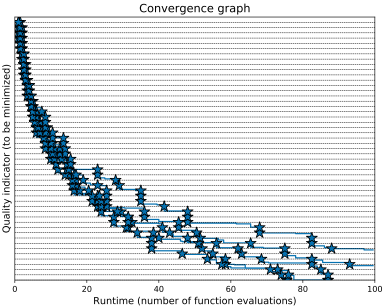
Data profile

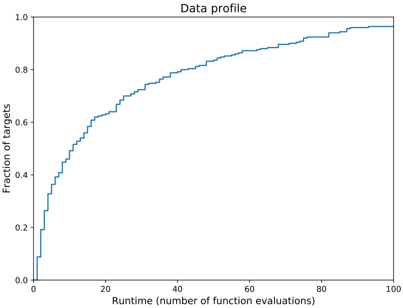
Data profiles can aggregate performance over multiple runs



Data profile

Data profiles can aggregate performance over multiple runs





COCO test suites

Test suites and algorithm results

- **`bbob`** test suite with 24 functions (173 algorithms)
- **`bbob-noisy`** test suite with 30 functions (45 algorithms)
- **`bbob-biobj`** test suite with 55 functions (16 algorithms)

Algorithm results collected at 9 BBOB Workshops (since 2009, mostly at GECCO conferences)

Under development

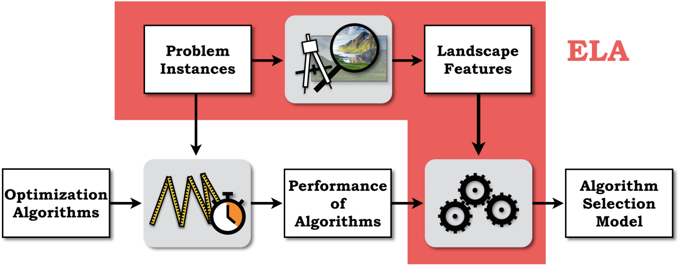
- Suite with constrained problems
- Suite with large-scale problems
- Suites with real-world problems

General Idea of Exploratory Landscape Analysis

Introduction

Goal:

- improve understanding of (continuous black-box) problems
- describe relationship between algorithm behavior and underlying problem
- ultimate goal for algorithm selection problem[1] (ASP): select the "best" algorithm



[1] Rice, J. (1976). *The Algorithm Selection Problem*. In: Advances in Computers (pp. 65 – 118).

Introduction

Idea of *Exploratory Landscape Analysis (ELA):*

- characterize black-box problems by numerical (and thus automatically computable) values
- start with very simple features without clear purpose
- match existing high-level features[2] with our ELA features

[2] high-level features = properties / characteristics of the problem landscape as categorized by an expert

Introduction

Notes I:

- functional relationships are unknown when designing features (usually one has a vague idea of what kind of property one would like to “measure”)
- pure numbers of a single feature on a single problem are basically meaningless
  - $\rightsquigarrow$ look at combination of features and/or compare the values across problems

Introduction

Notes II:

- try to match the features to high-level characteristics[5] (multimodality, funnel structure, etc.) of optimization problems
- this enables recognizing important problem properties quickly (and without consulting an expert)

[5] usually via classification models, whose "class labels" are the problem properties

Introduction

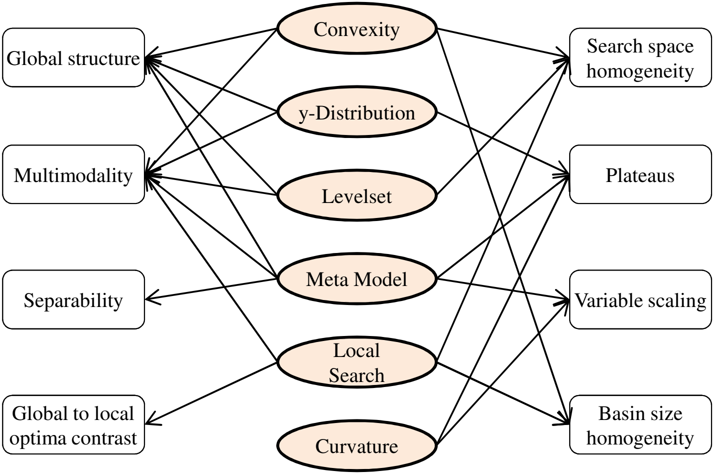
Notes III:

- features are based on initial design of samples $x_{i1}, \ldots, x_{iD}$ and their corresponding fitness values y_i, $i = 1, \ldots, n$
- given an evaluated initial design[6], most ELA features are for free $\rightsquigarrow$ they don't need any further function evaluations
- multiple different feature sets already exist, and we will introduce some of them on the following slides[7]

[6] usually a well-spread sample (LHS, random uniform sample, etc.); however, using the initial population of an optimizer is also possible

[7] for further details, please attend "ELA Tutorial" at PPSN 2018 ;-)

Introduction



Mersmann, O., Preuss, M. & Trautmann, H. (2010). *Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis*. In: Proceedings of PPSN XI (pp. 71 - 80).

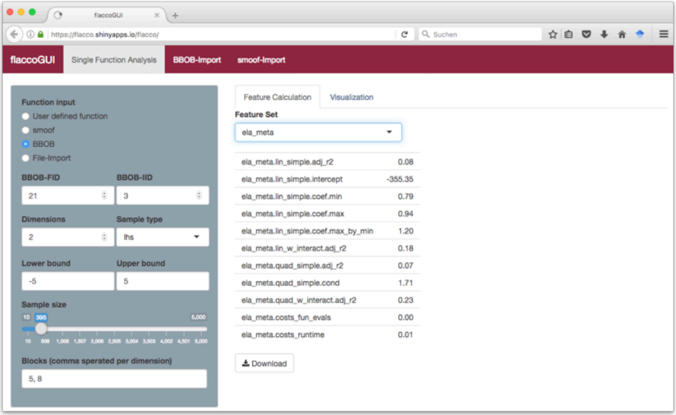
FLACCO + GUI

Notes I:

- `flacco`: **F**eature-Based **L**andscape **A**nalysis of **C**ontinuous and **C**onstraint **O**ptimization Problems
- unified interface for multiple (single-objective) sets of configurable features
- stable release on CRAN / developers version on GitHub
- multiple vizualisation techniques (partially shown on these slides)

Notes II:

- `flacco` also comes with a platform-independent web-application



[8]

[8] Link to GUI: `https://flacco.shinyapps.io/flacco/`

Notes III:

- tracks # of function evaluations and run time - per feature set
- FLACCO is described in our CEC paper:

  Kerschke, P. & Trautmann, H. (2016). *The R-Package FLACCO for Exploratory Landscape Analysis with Applications to Multi-Objective Optimization Problems*. In: Proceedings of CEC 2016.

- further information on FLACCO, its GUI, or the contained feature sets can be found here:

  Kerschke, P. (2017). *Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package flacco*. In: https://arxiv.org/abs/1708.05258.

Table of Contents

Top Trumps: Rules

1: Shuffle deck and distribute evenly among players
2: Starting player chooses characteristic (category)
3: All players compare corresponding values on their cards
4: Player with *highest* value wins trick
5: ~~Until at least one player has lost all their cards~~
5: Until at all cards have been played exactly once
6: Winning player announces new characteristic, goto 3

Fitness Functions

Agents

- both remember all previously played cards

KA **K**nowledgable **A**gent: Knows the exact values of all cards in the deck

NA **N**aïve **A**gent: Only knows the valid value ranges

id	name	description	range
1	deckHV	deck hypervolume maximising card values	[0,?]
2	catSD	standard deviation of category means	[0,?]
3	fair	KA (Knowledgable player) winrate	[0,1]
4	leadChange	average # trick changes	[0,16]
5	trickDiff	average trick difference	[0,16]

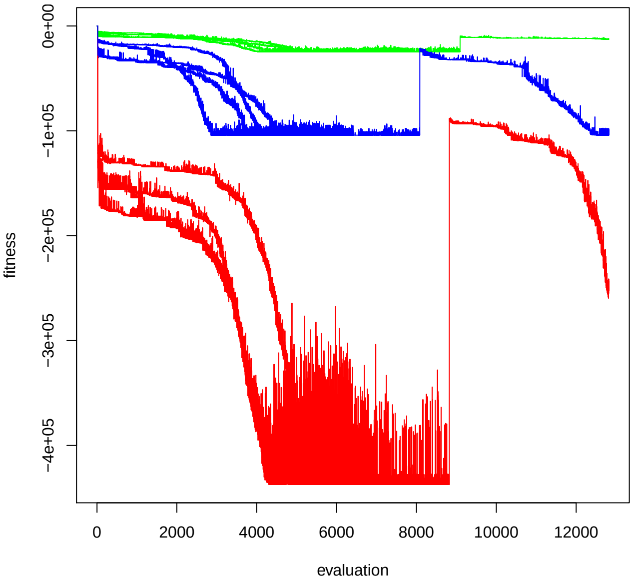
Instances

32 cards, 4 categories ⇒ dimension 128

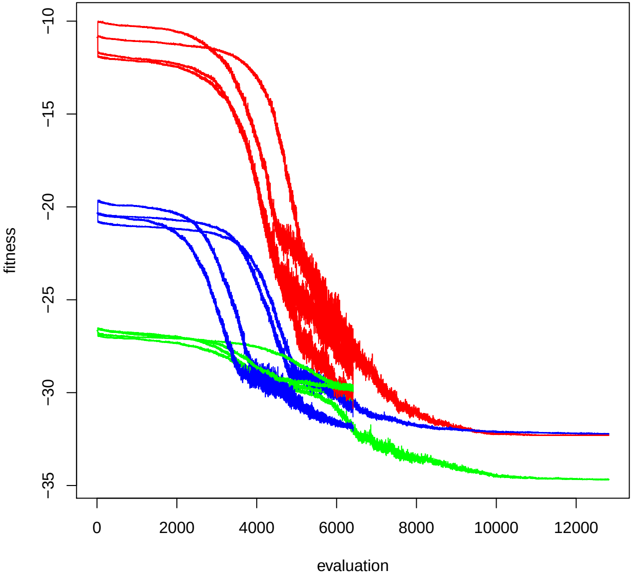
Category bounds

1. Instance 1: [39, 84] x [78, 80] x [20, 91] x [34, 77]
2. Instance 2: [70, 81] x [09, 12] x [35, 42] x [07, 70]
3. Instance 3: [22, 56] x [39, 44] x [14, 29] x [56, 86]

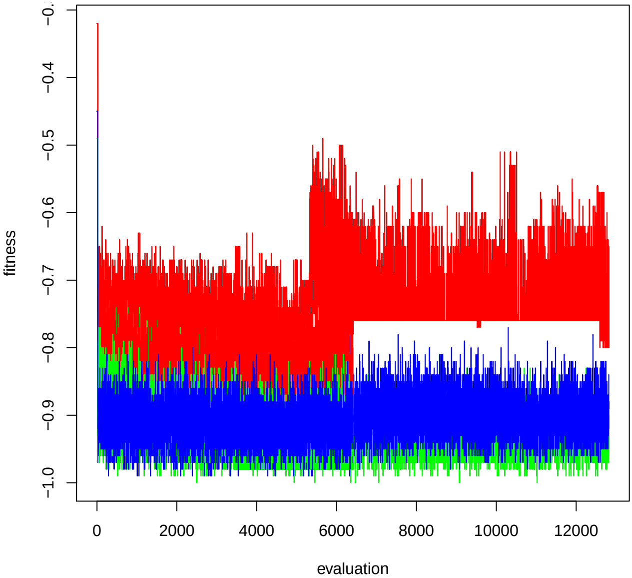
CMA-ES Performance: deckHV, dim 128, [0,?]



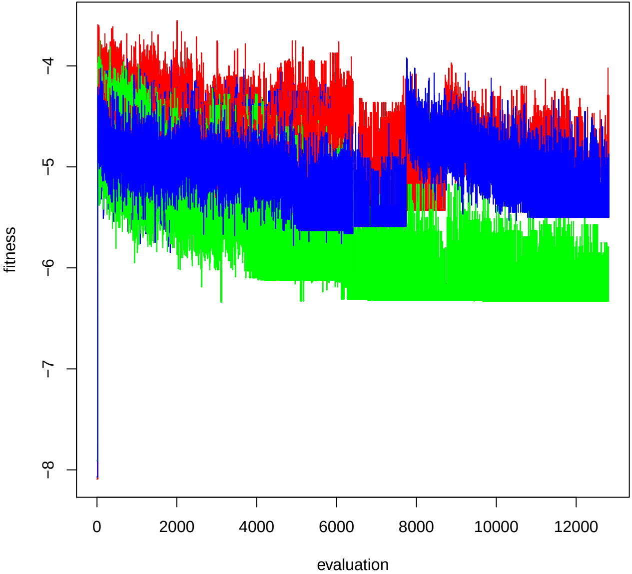
CMA-ES Performance: catSD, dim 128, [0,?]



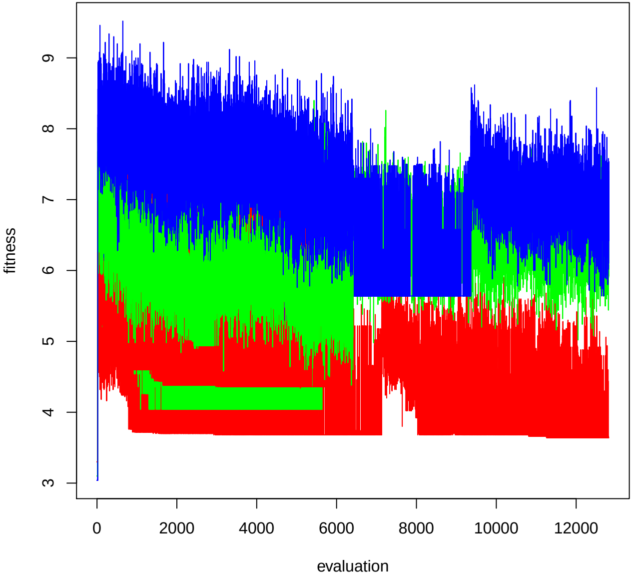
CMA-ES Performance: fair, dim 128, [0,1]



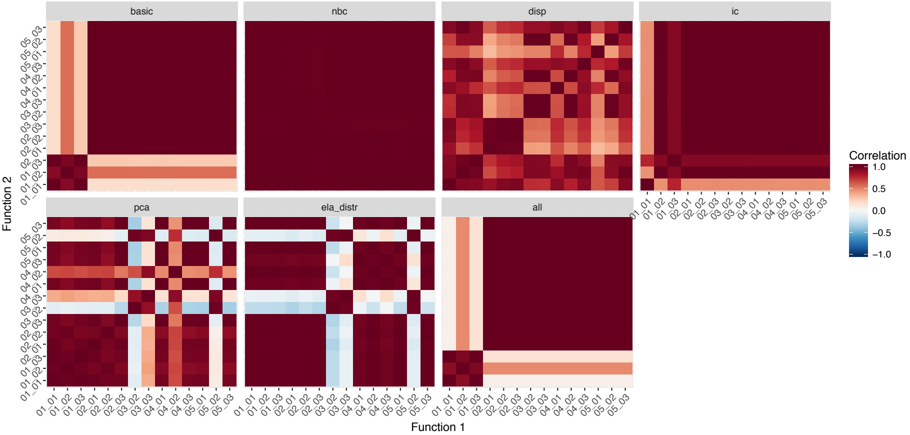
CMA-ES Performance: leadChange, dim 128, [0,16]



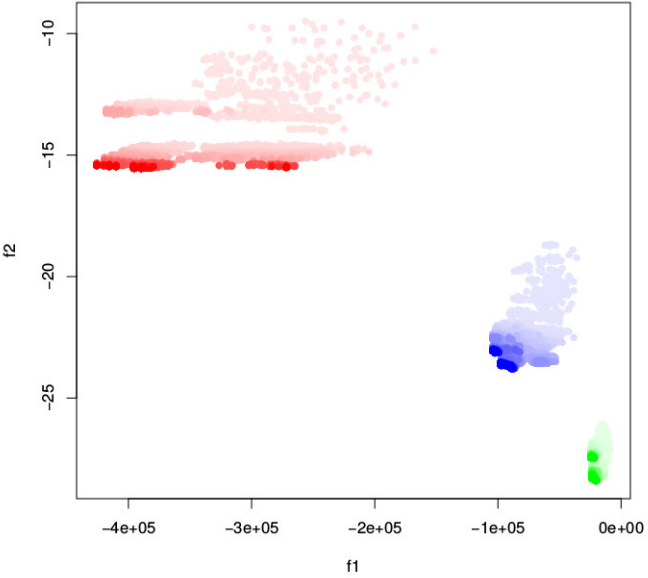
CMA-ES Performance: trickDiff, dim 128, [0,16]



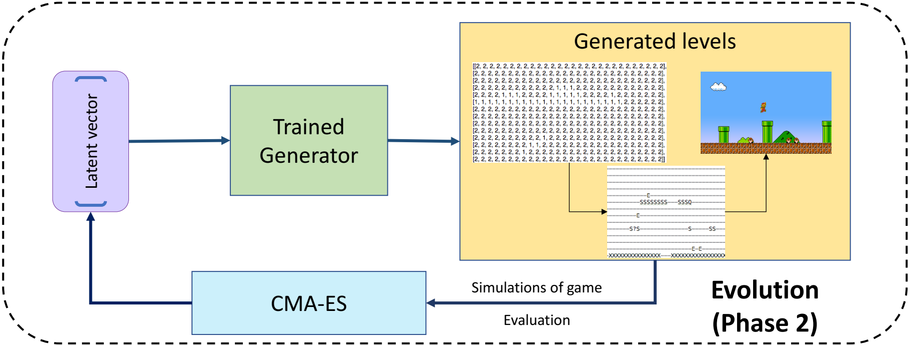
Results: ELA on TopTrumps



SMS-EMOA Performance: deckHV vs. catSD



Procedural Level Generator for Mario



Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. Lucas, Adam Smith, Sebastian Risi. 2018. Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. In Genetic and Evolutionary Computation Conference (GECCO 2018). ACM Press, New York, NY. To appear.

Example

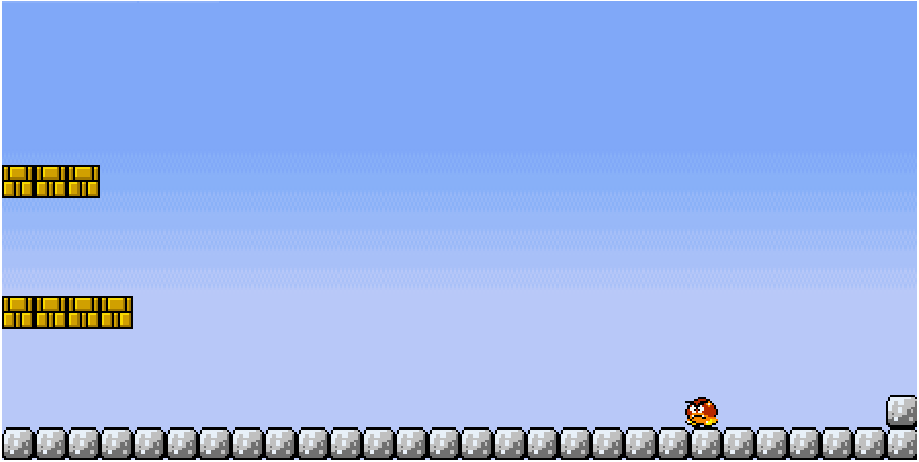
Latent Vector

```
[0.37096528435428605, 0.4875451956823884, 0.5442587474115113,
-0.4297413700372004, -0.17310705605523974, 0.15561409410805174,
0.30666730352848920, 0.10269919817016136, 0.08195305887271840,
-0.6667159059020512]
```

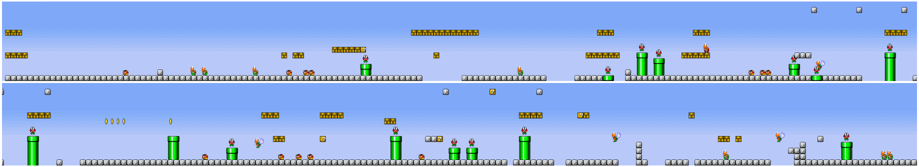
GAN output

```
[[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 10, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 9, 2, 2, 2, 2, 2, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

Example cont'd



In action



Fitness Functions, Dimensions and Instances

Trained GANs

- latent vector dimensions: 10, 20, 30, 40
- output dimension: 28 x 14
- sample sets:
  - Super Mario Bros: overworld lvls
  - Super Mario Bros: underground lvls
  - Super Mario Bros: overworld lvls + Super Mario Bros 2 (Japan): overworld lvls
- Random seed (instances)

Fitness Functions

- 6 direct fitness functions*
- 4 simulated: AStar Agent and REALM†
- Concatenation

*Adam Summerville, Julian R. H. Mariño, Sam Snodgrass, Santiago Ontañón, Levi H. S. Lelis. 2017. Understanding mario: an evaluation of design metrics for platformers. In Foundations of Digital Games (FDG 2017). ACM Press, New York, NY. 8:1-8:10.

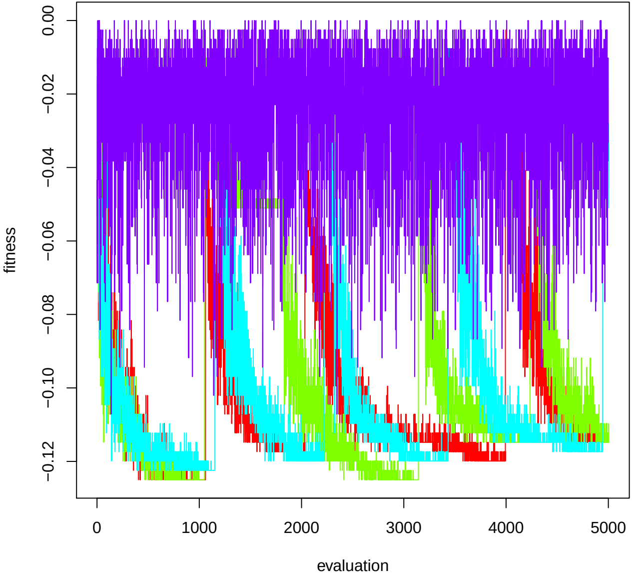
†Agents by R. Baumgarten and S. Bojarski, C. B. Congdon, MarioAI Competition

Selected Fitness Functions

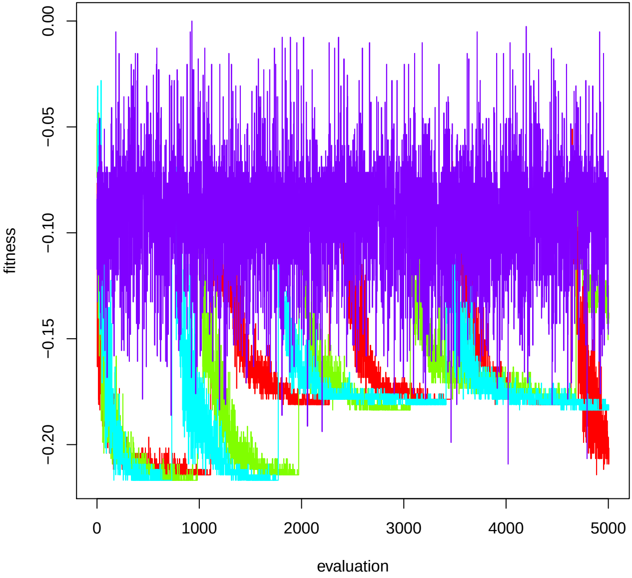
id	name	description	range
9	decorationPerc	percentage of *pretty* tiles	[0,1]
12	negativeSpace	percentage of tiles you can stand on	[0,1]

id	name	description	range
21 / 33	levelProgress	level progress x-wise	[0,1]
24 / 36	basicFitness	lengthOfLevelPassedPhys - timeSpentOnLevel + numberOfGainedCoins + marioStatus*5000)/5000	?
27 / 39	jumpFraction	percentage of jump actions	[0,1]
30 / 42	totalActions	number of actions total	[0,?]

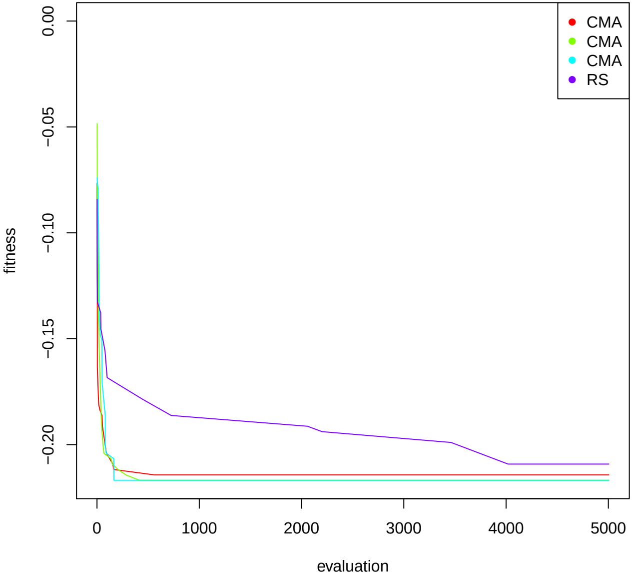
Algorithm Performance: decorationPerc, dim 10, [0,1]



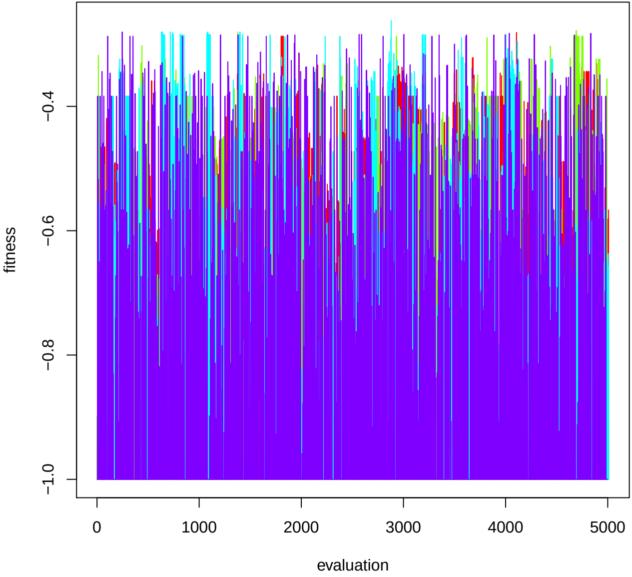
Algorithm Performance: negativeSpace, dim 10, [0,1]



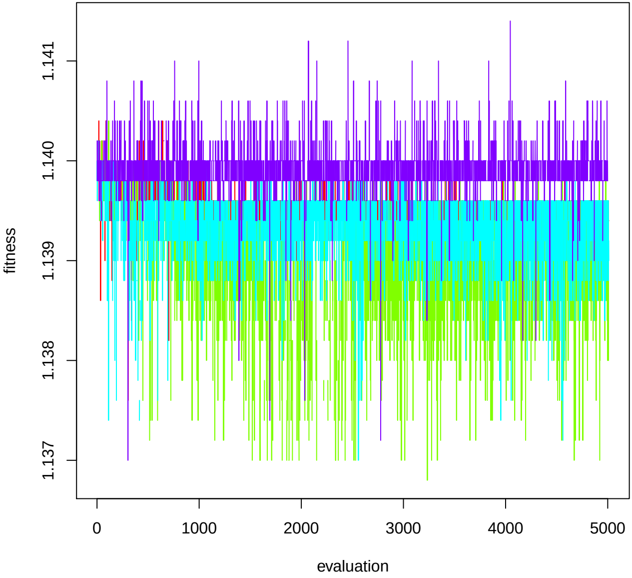
Algorithm Performance: negativeSpace, dim 10, [0,1]



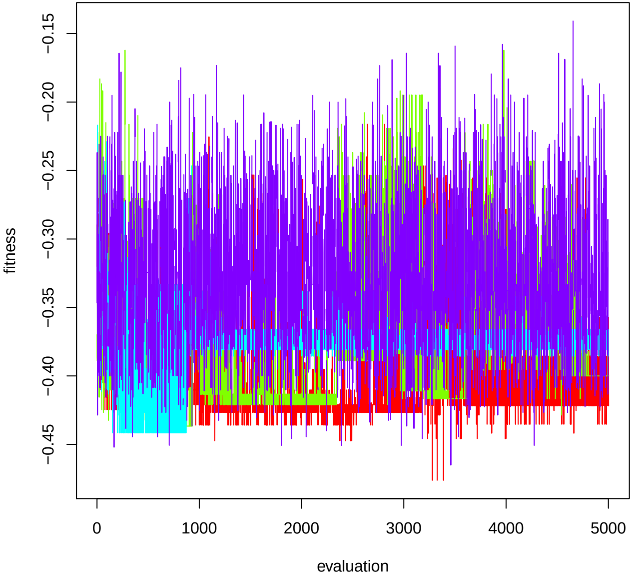
Algorithm Performance: levelProgress AStar, dim 10, [0,1]



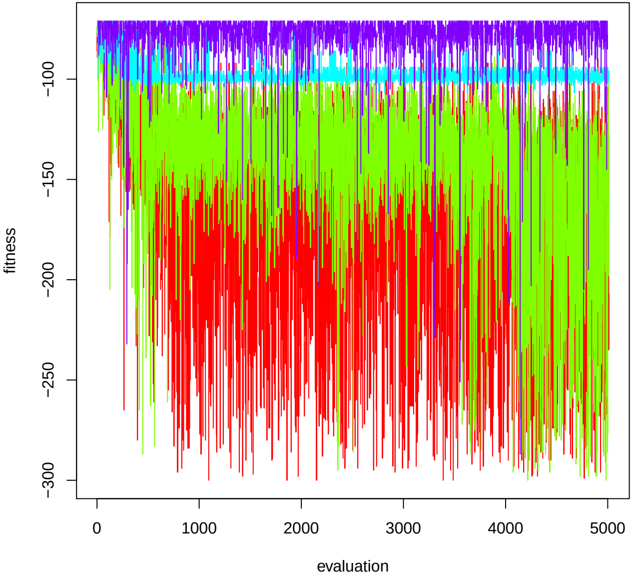
Algorithm Performance: basicFitness AStar, dim 10, [0,1]



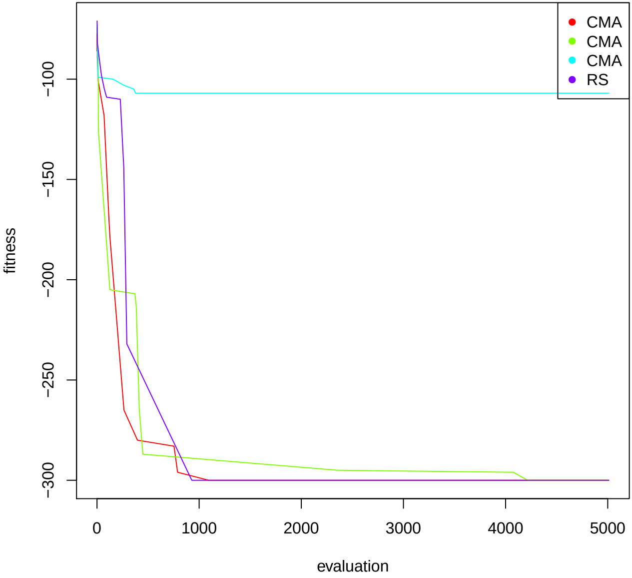
Algorithm Performance: jumpFraction AStar, dim 10, [0,1]



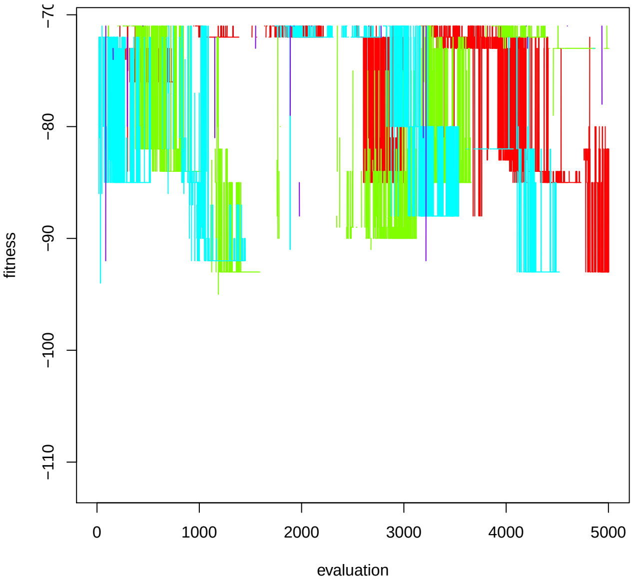
Algorithm Performance: totalActions AStar, dim 10, [0,1]



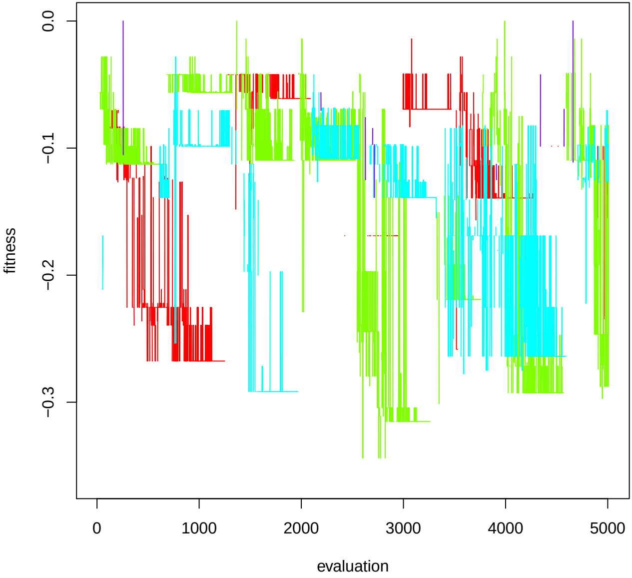
Algorithm Performance: totalActions AStar, dim 10, [0,1]



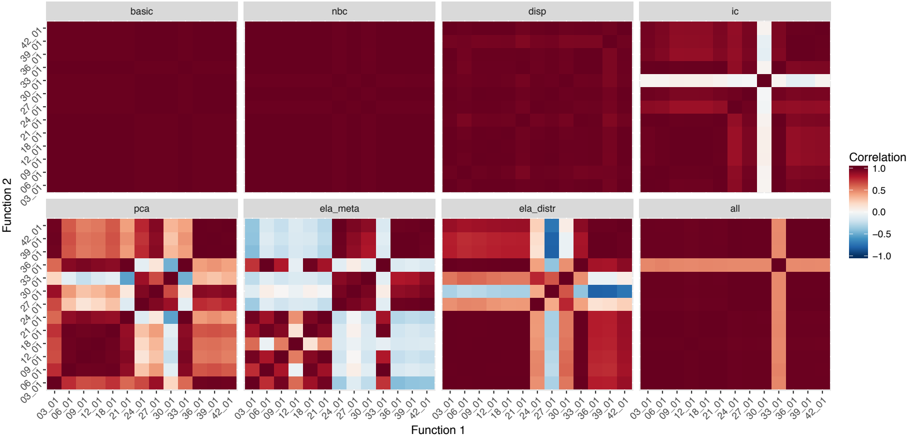
Algorithm Perf.: totalActions REALM, dim 10, [0,1]



Algorithm Perf.: jumpFractions REALM, dim 10, [0,1]



Results: ELA on MarioGAN CMA



Results: ELA on MarioGAN RS

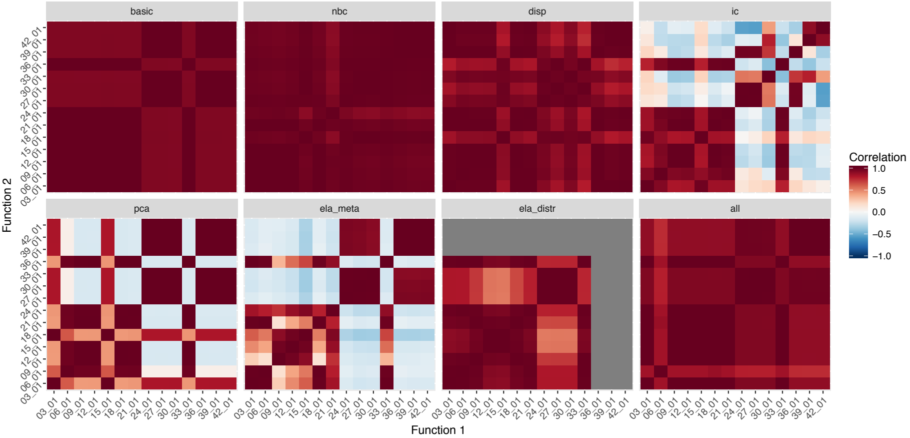


Table of Contents

Topics

Benchmark Requirements: EC Perspective

- Suitability of fitness functions (e.g. too easy, no correlation)
- Interesting characteristics

Benchmark Requirements: Games Perspective

- Representative fitness functions $\Rightarrow$ Generalisability
- Sensibility of fitness functions (e.g. enemy distribution)
- Interesting characteristics

Analysis

- Suitable measures and approaches to analyse fitness landscapes
- Suggestions for choice of algorithm
- Representations that simplify landscapes
- Noise in stochastic simulations

Considered ELA Features

Considered ELA Features

Meta-Model Features:

- fits linear and quadratic models (with and without pairwise interaction effects) to the data
- extracts information from these models, such as ...
  - ... the adjusted R^2 of these models
  - ... summary statistics of the estimated parameter coefficients
- helpful to ...
  - ... detect simple problems such as 'sphere' or 'linear slope'
  - ... distinguish between problems with an underlying global structure (e.g., funnel) and random landscapes

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C. & Rudolph, G. (2011). *Exploratory Landscape Analysis*. In: Proceedings of GECCO 2011 (pp. 829 – 836)

Considered ELA Features

y-Distribution Features:

- focusses on distribution of objective values ($= y$-values)
- measures skewness, kurtosis and (estimated) number of peaks of the distribution of the y-values
- helpful to detect, whether landscape possesses many points at a certain height
  $\rightsquigarrow$ possible plateaus, mainly flat areas, spiky peaks, ...?

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C. & Rudolph, G. (2011). *Exploratory Landscape Analysis*. In: Proceedings of GECCO 2011 (pp. 829 – 836)

Considered ELA Features

Dispersion Features:

- splits data based on a quantile of the objective values (default: best 2, 5, 10 and 25% vs. corresponding worst)
- computes average distance (mean and median) within group of worst and best observations $\rightsquigarrow$ aggregate via ratio or difference
- helpful to distinguish highly multimodal problems (with random global structure) from funnel-like (or other simpler) landscapes

Lunacek, M. & Whitley, D. (2006). *The Dispersion Metric and the CMA Evolution Strategy*. In: Proceedings of GECCO 2006 (pp. 477 - 484).

Considered ELA Features

Nearest Better Clustering Features:

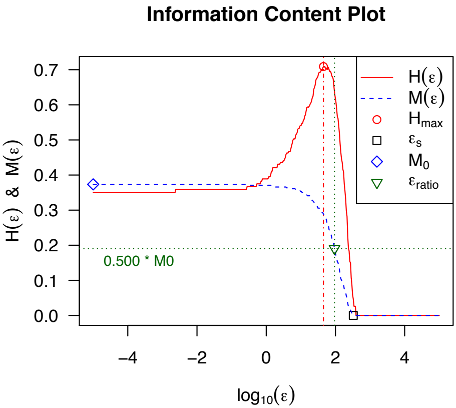
- computes for each observation the nearest neighbor and nearest better neighbor (= closest neighbor among all observation with better y-value)
- analyze the two distance sets (set of nearest neighbor distances and set of nearest better neighbor distances)
- proved to be helpful for detecting funnel landscapes

Kerschke, P., Preuss, M., Wessing, S. & Trautmann H. (2015). *Detecting Funnel Structures by Means of Exploratory Landscape Analysis*. In: Proceedings of GECCO 2015 (pp. 265 - 272).

Considered ELA Features

Information Content Features:

- based on a random walk along the sample's points
- aggregates information of changes (decrease, increase) for consecutive points along that walk
- helpful to 'measure' smoothness, ruggedness, or neutrality of a landscape



Muñoz, M. A., Kirley, M., Halgamuge, S. K. (2015). *Exploratory Landscape Analysis of Continuous Space Optimization Problems using Information Content*. In: IEEE Transactions on Evolutionary Computation (pp. 74 - 87).

Considered ELA Features

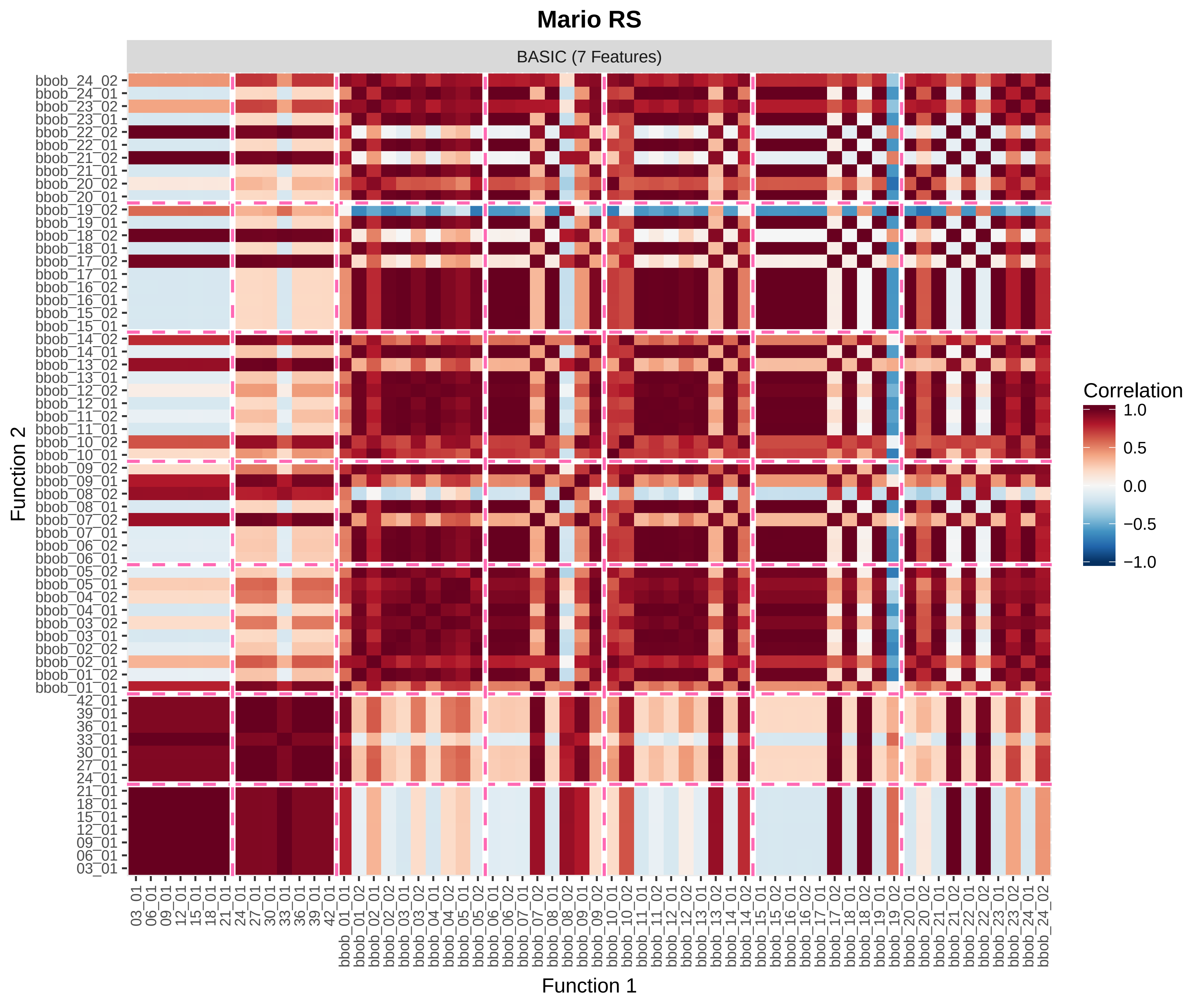
Basic Features:

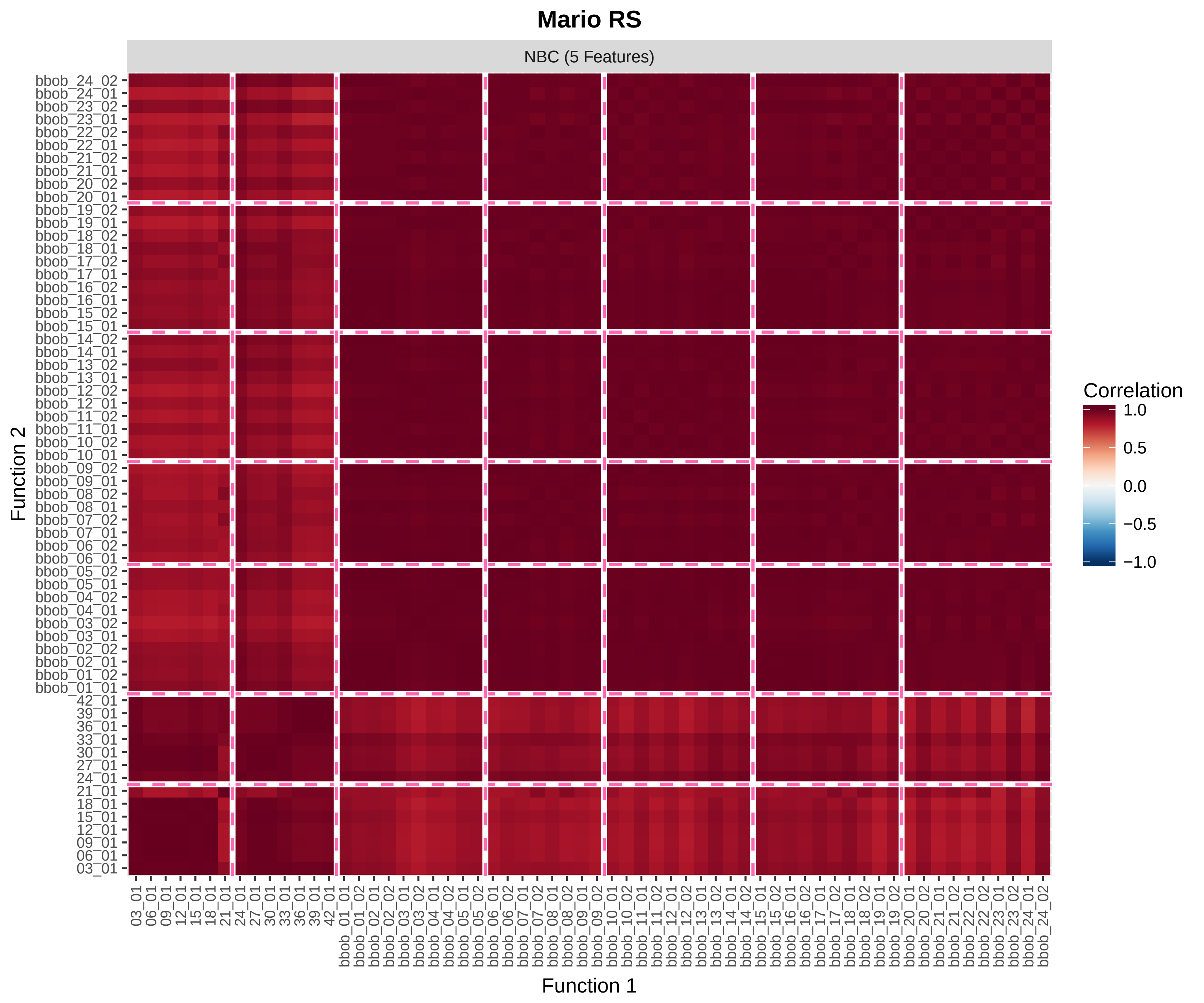
- straight-forward information from the problem setup, such as number of input parameters, observations, boundaries, etc.

Principal Component Analysis Features:

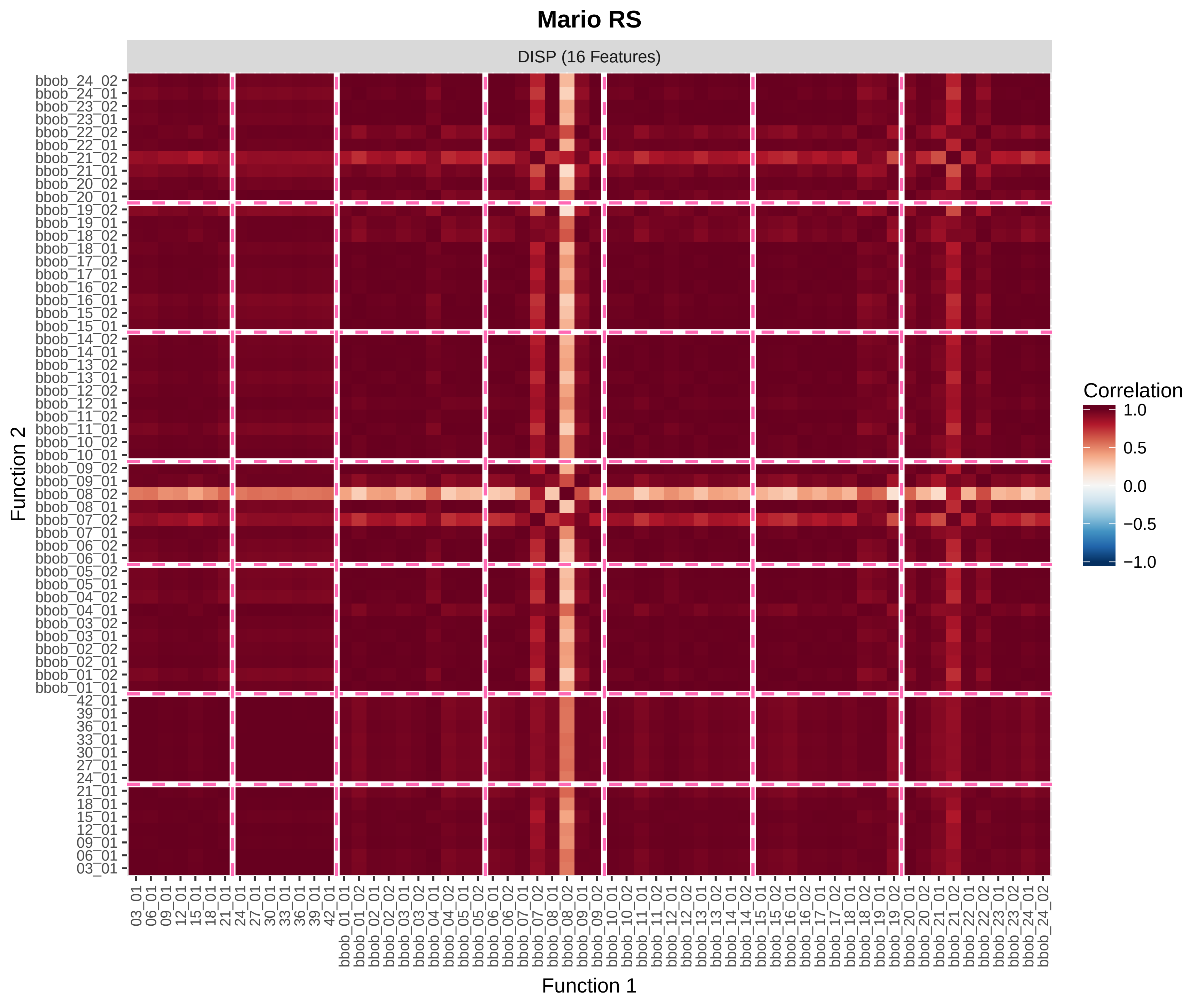
- information based on applying PCA ($\rightsquigarrow$ dimensionality reduction) on the landscape, e.g., percentage of variance that is explained by the first principal component

Kerschke, P. (2017). *Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package flacco*.
In: `https://arxiv.org/abs/1708.05258`.

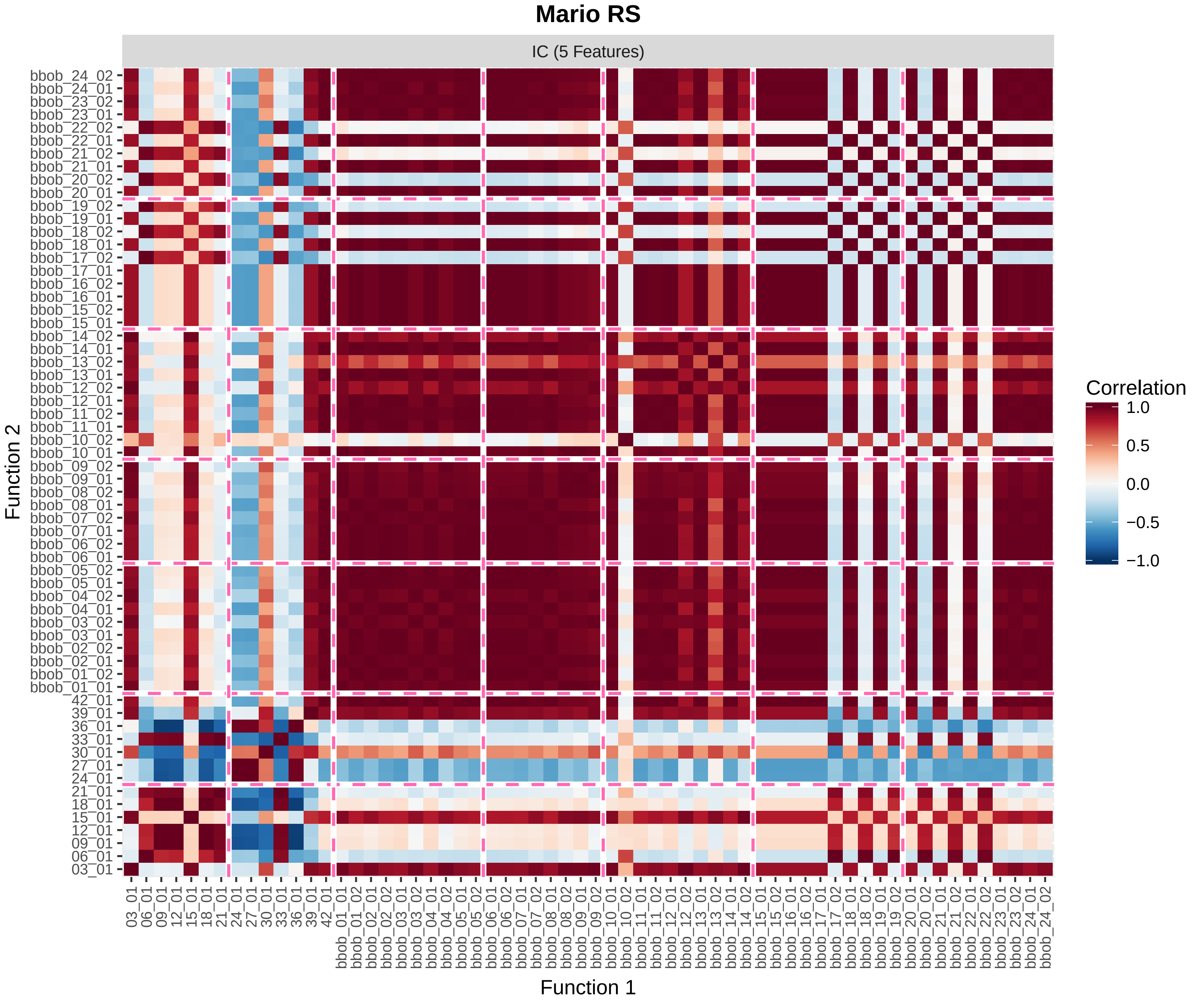

Mario RS
BASIC (7 Features)
Correlation
1.0
0.5
0.0
−0.5
−1.0
Function 1
Function 2



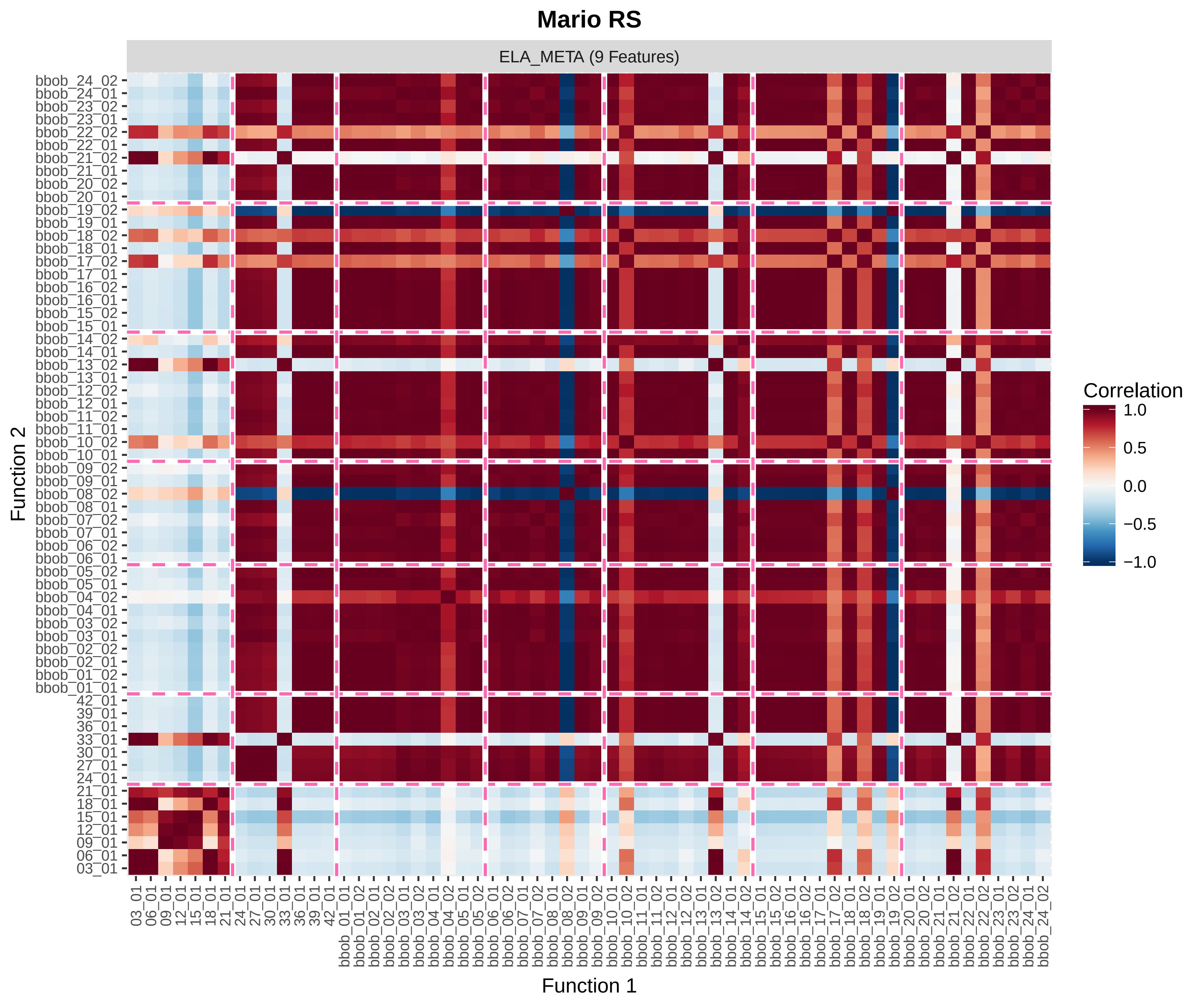
Mario RS
NBC (5 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0

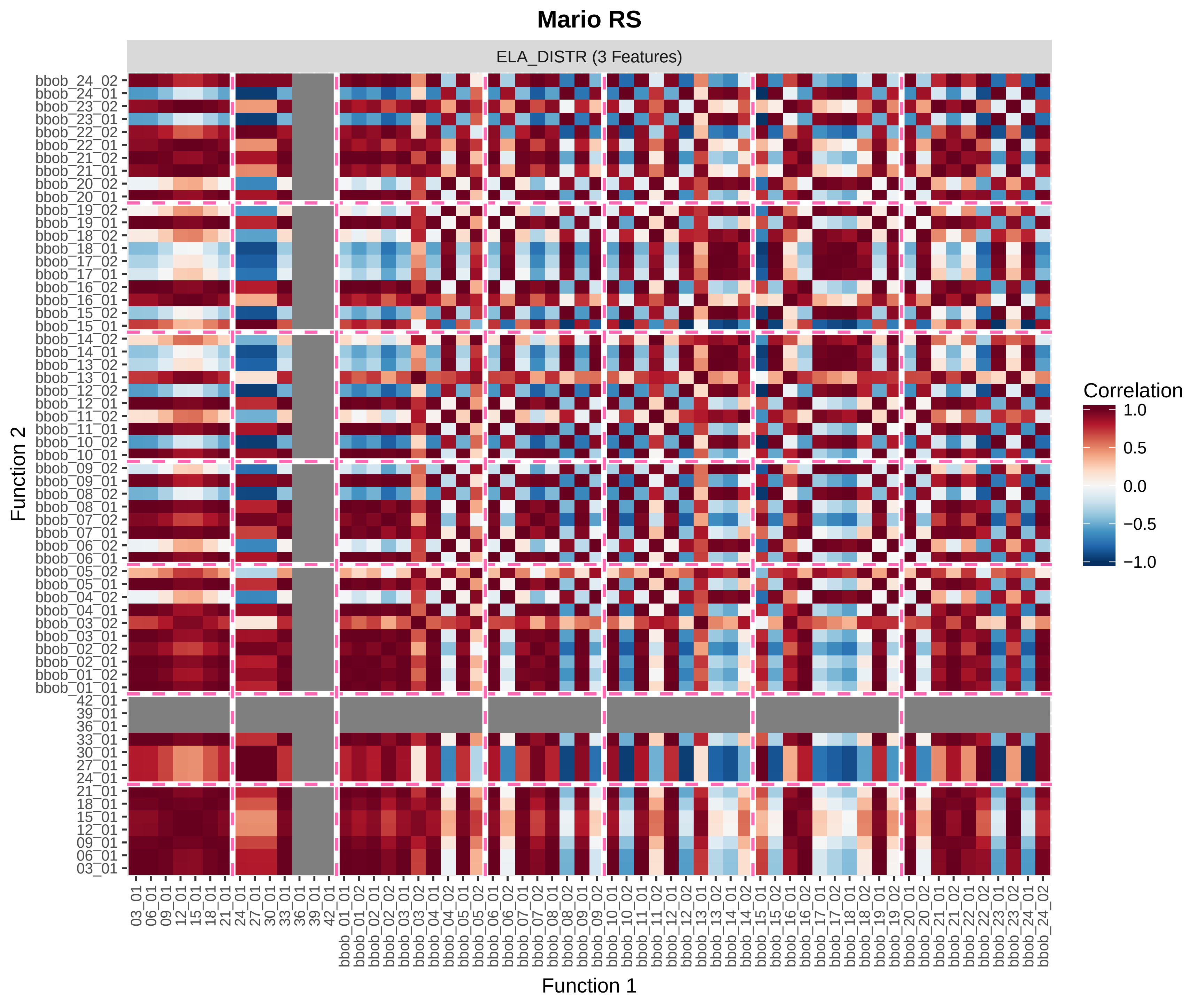


Mario RS
DISP (16 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0

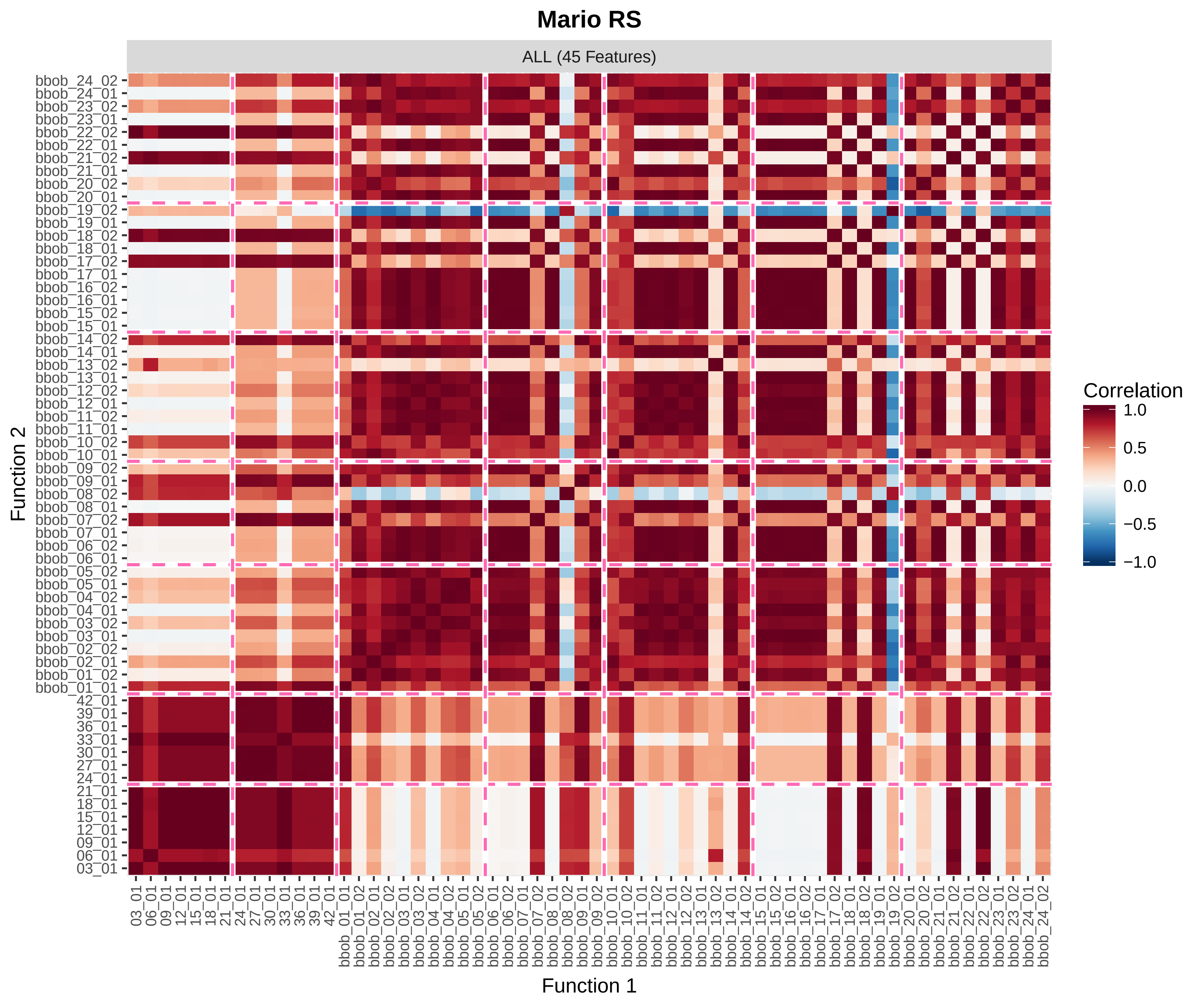


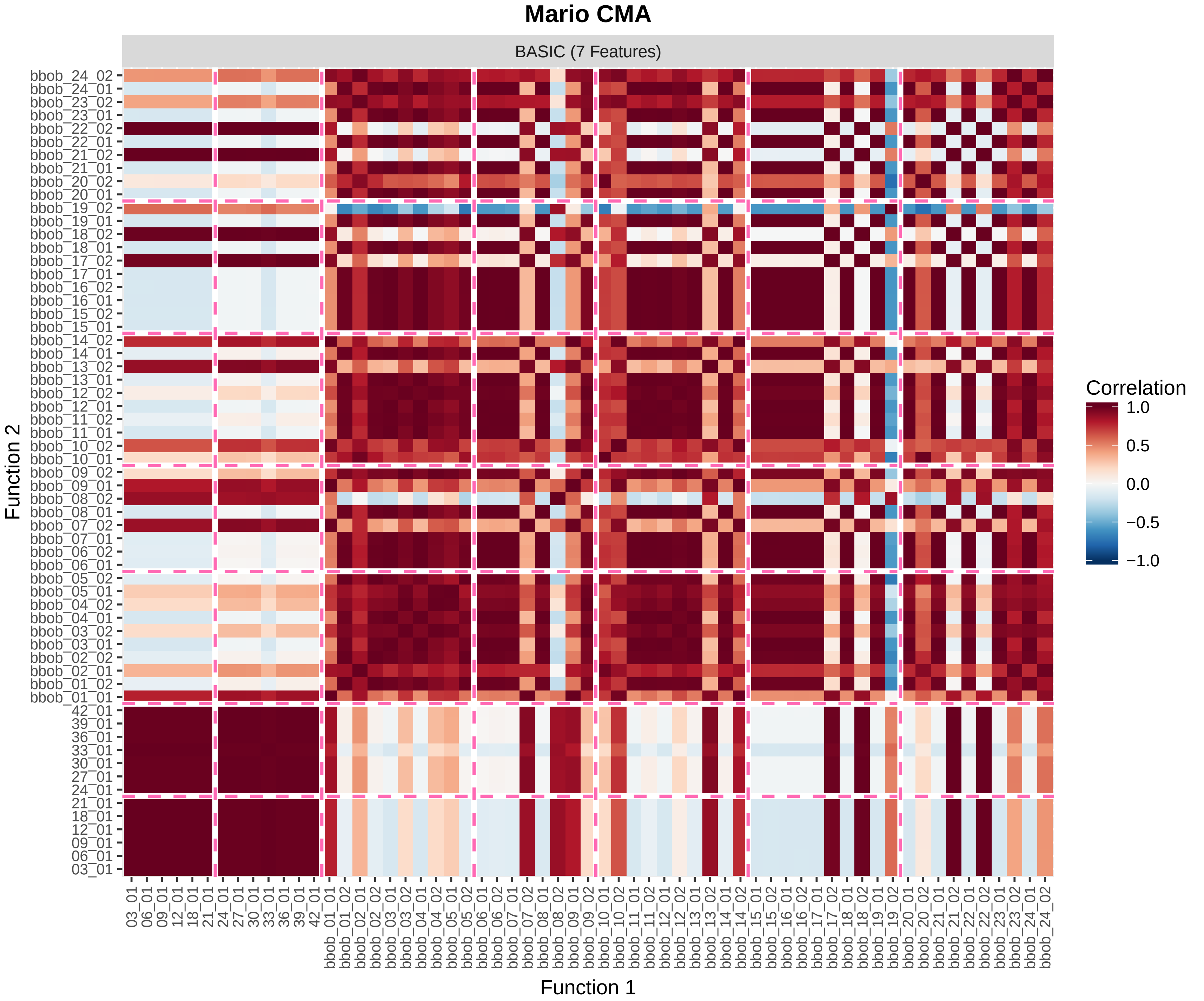
Mario RS
IC (5 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0

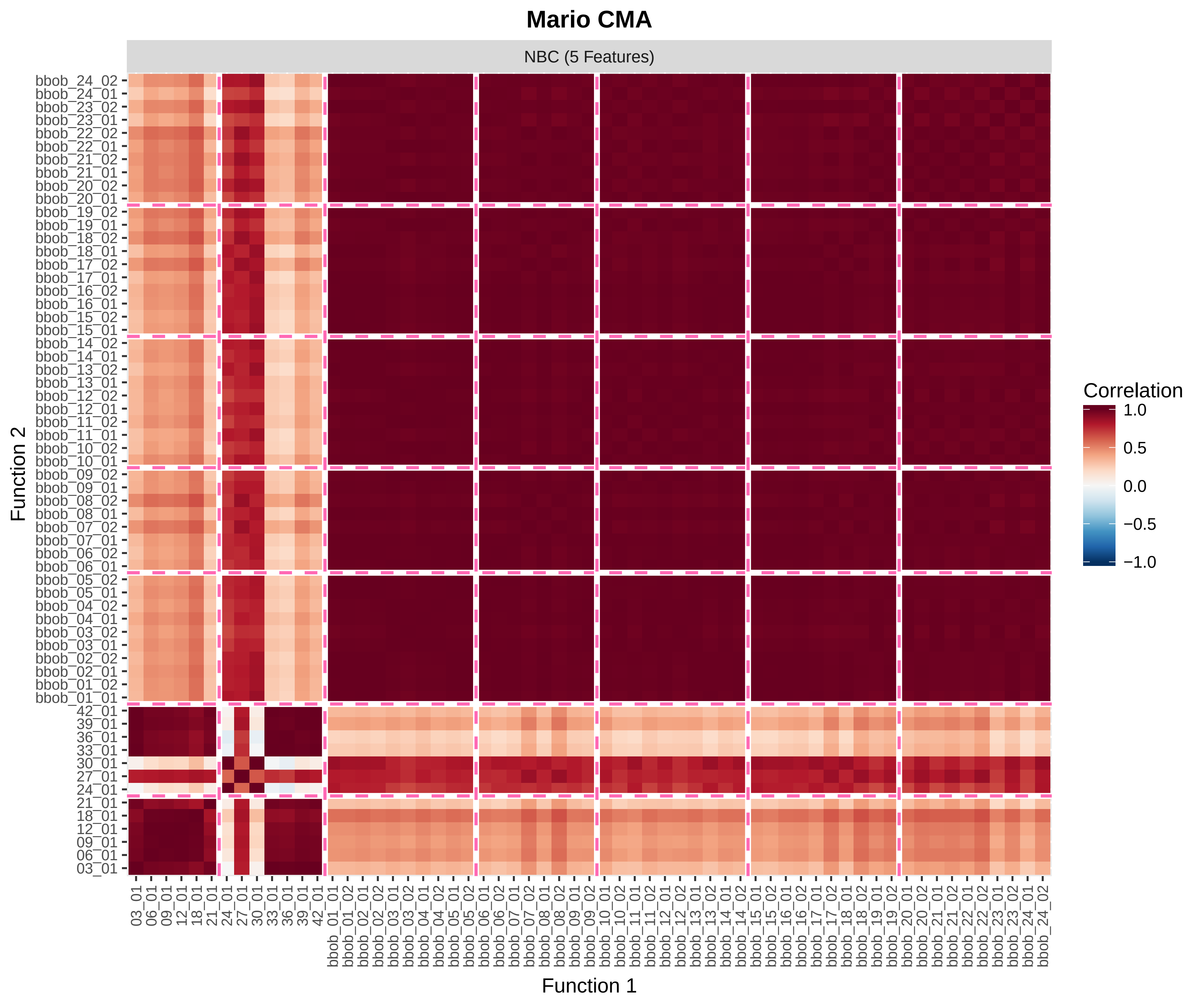

Mario RS
ELA_META (9 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0



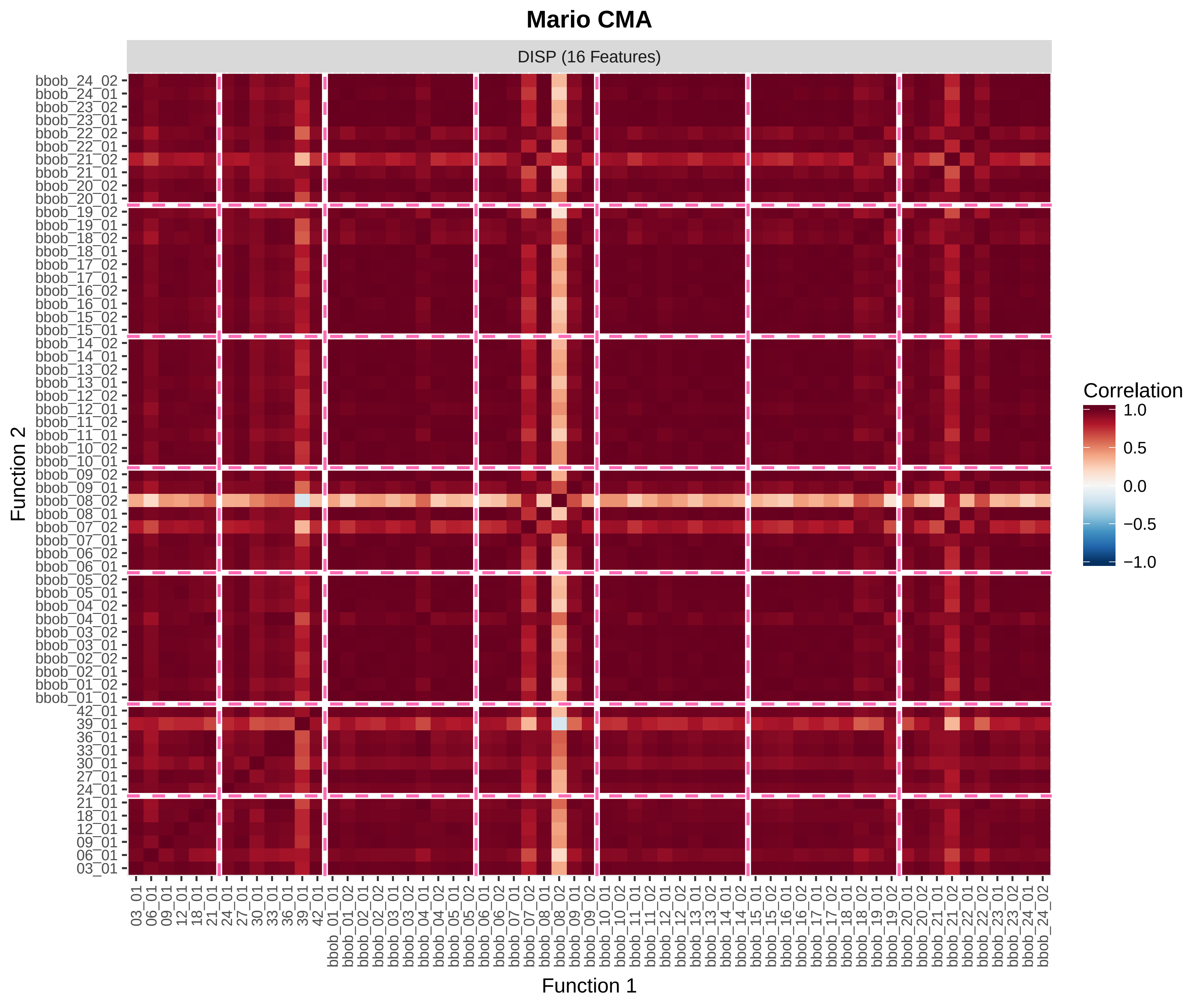
Mario RS
ELA_DISTR (3 Features)
Function 1
Function 2
Correlation
1.0
0.5
0.0
−0.5
−1.0

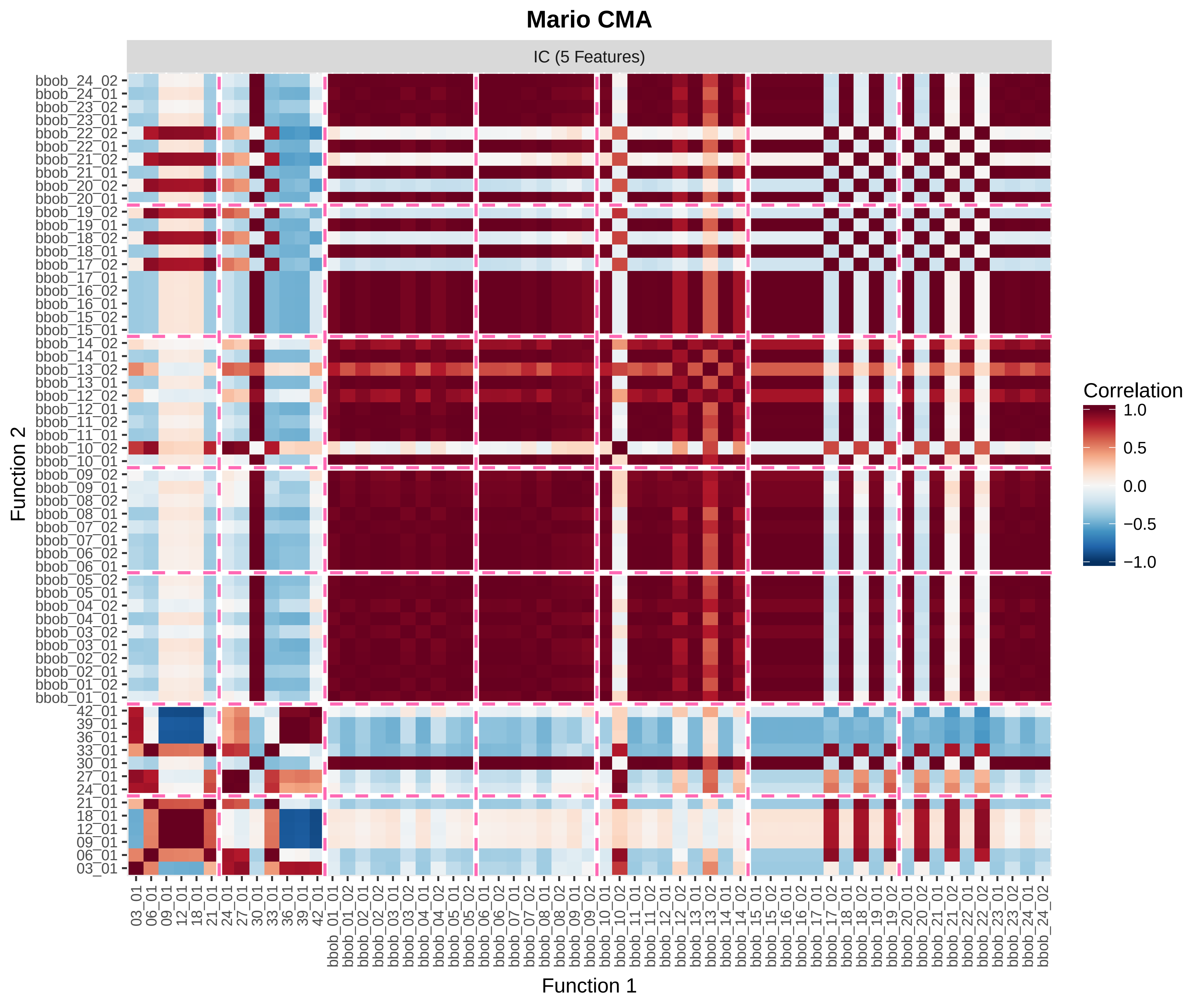

Mario RS
ALL (45 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0
bbob_24_02
bbob_24_01
bbob_23_02
bbob_23_01
bbob_22_02
bbob_22_01
bbob_21_02
bbob_21_01
bbob_20_02
bbob_20_01
bbob_19_02
bbob_19_01
bbob_18_02
bbob_18_01
bbob_17_02
bbob_17_01
bbob_16_02
bbob_16_01
bbob_15_02
bbob_15_01
bbob_14_02
bbob_14_01
bbob_13_02
bbob_13_01
bbob_12_02
bbob_12_01
bbob_11_02
bbob_11_01
bbob_10_02
bbob_10_01
bbob_09_02
bbob_09_01
bbob_08_02
bbob_08_01
bbob_07_02
bbob_07_01
bbob_06_02
bbob_06_01
bbob_05_02
bbob_05_01
bbob_04_02
bbob_04_01
bbob_03_02
bbob_03_01
bbob_02_02
bbob_02_01
bbob_01_02
bbob_01_01
42_01
39_01
36_01
33_01
30_01
27_01
24_01
21_01
18_01
15_01
12_01
09_01
06_01
03_01


Mario CMA
BASIC (7 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0

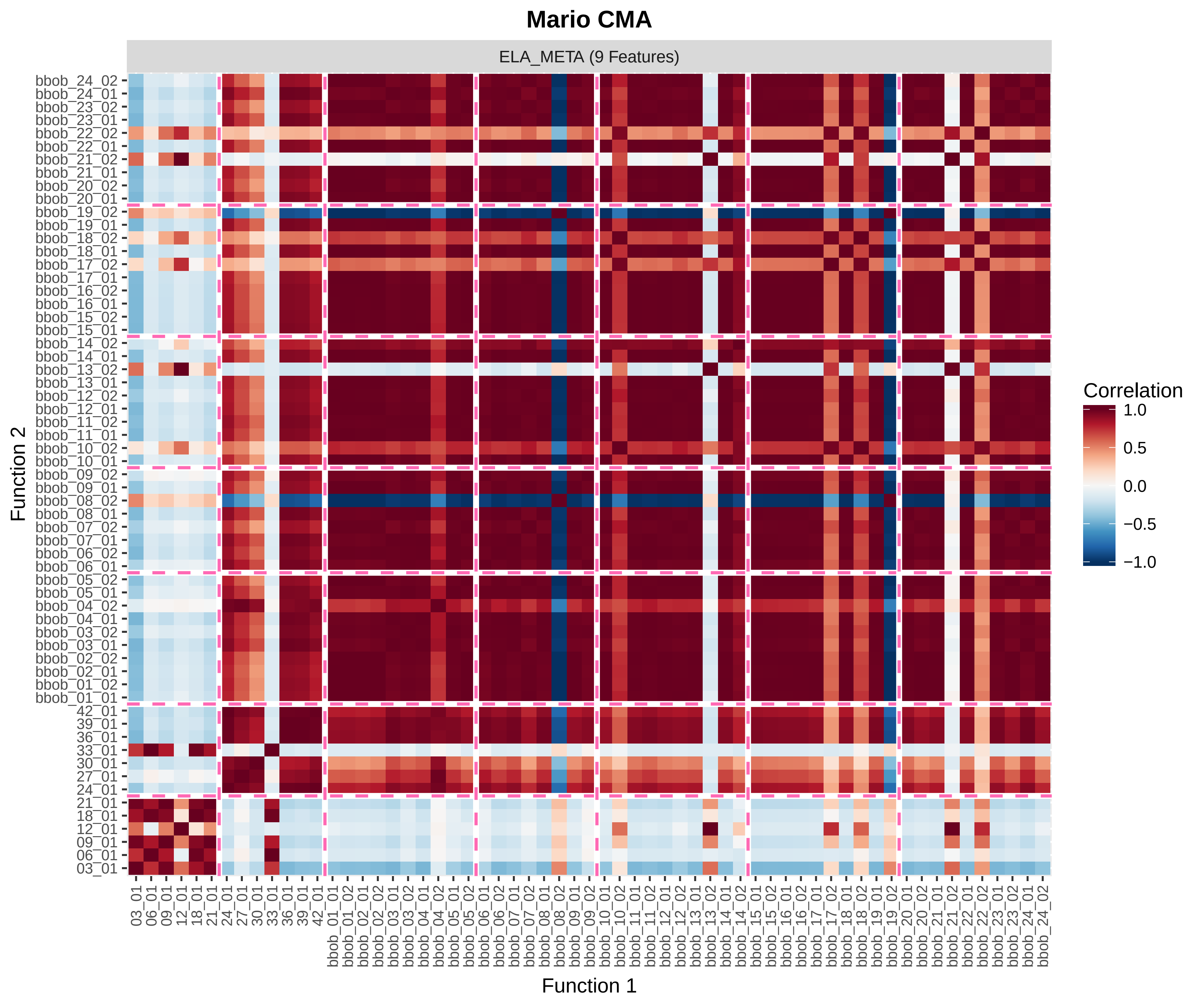


Mario CMA
NBC (5 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0
bbob_24_02
bbob_24_01
bbob_23_02
bbob_23_01
bbob_22_02
bbob_22_01
bbob_21_02
bbob_21_01
bbob_20_02
bbob_20_01
bbob_19_02
bbob_19_01
bbob_18_02
bbob_18_01
bbob_17_02
bbob_17_01
bbob_16_02
bbob_16_01
bbob_15_02
bbob_15_01
bbob_14_02
bbob_14_01
bbob_13_02
bbob_13_01
bbob_12_02
bbob_12_01
bbob_11_02
bbob_11_01
bbob_10_02
bbob_10_01
bbob_09_02
bbob_09_01
bbob_08_02
bbob_08_01
bbob_07_02
bbob_07_01
bbob_06_02
bbob_06_01
bbob_05_02
bbob_05_01
bbob_04_02
bbob_04_01
bbob_03_02
bbob_03_01
bbob_02_02
bbob_02_01
bbob_01_02
bbob_01_01
42_01
39_01
36_01
33_01
30_01
27_01
24_01
21_01
18_01
12_01
09_01
06_01
03_01

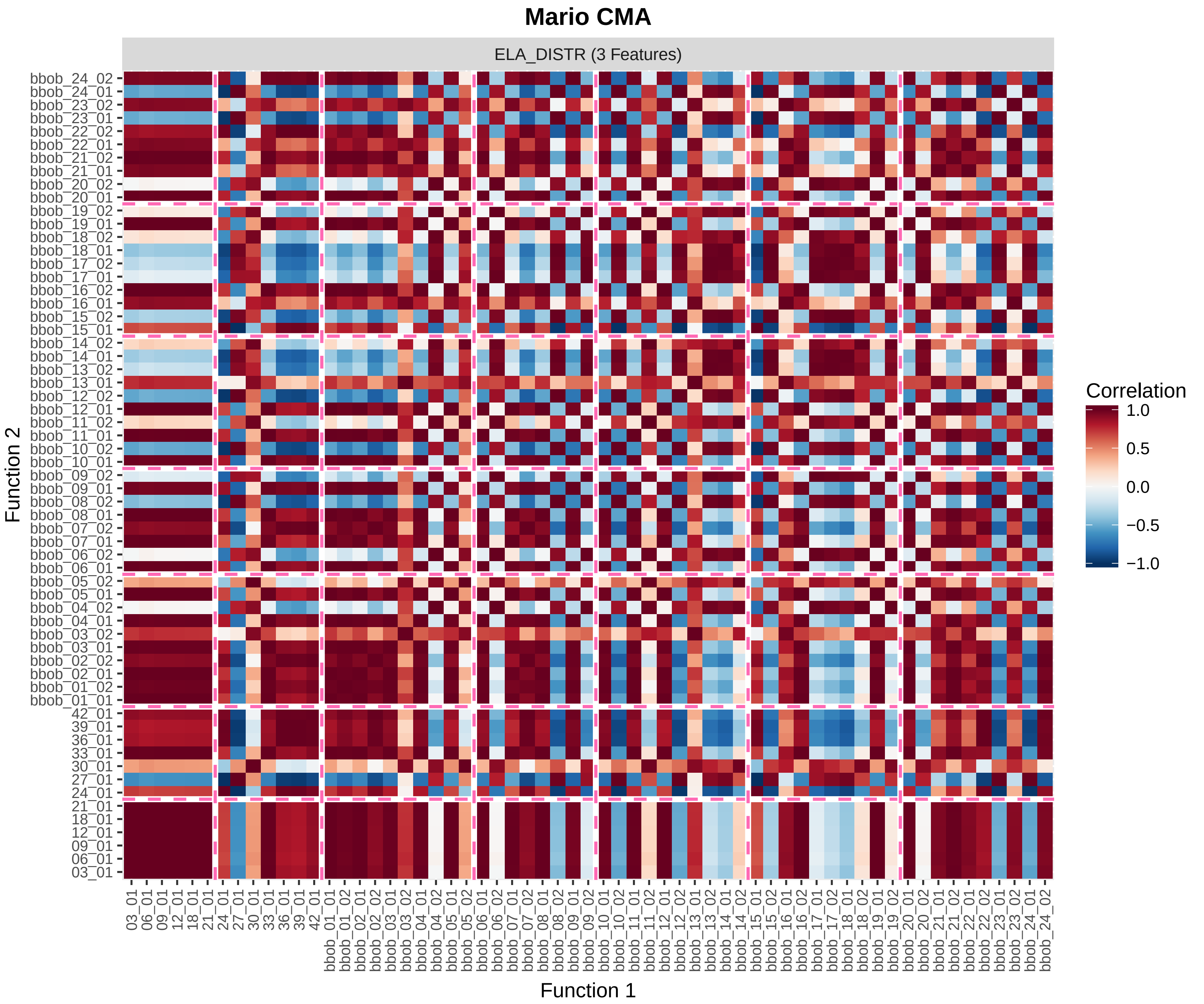

Mario CMA
DISP (16 Features)
Correlation
1.0
0.5
0.0
−0.5
−1.0
Function 1
Function 2
03_01
06_01
09_01
12_01
18_01
21_01
24_01
27_01
30_01
33_01
36_01
39_01
42_01
bbob_01_01
bbob_01_02
bbob_02_01
bbob_02_02
bbob_03_01
bbob_03_02
bbob_04_01
bbob_04_02
bbob_05_01
bbob_05_02
bbob_06_01
bbob_06_02
bbob_07_01
bbob_07_02
bbob_08_01
bbob_08_02
bbob_09_01
bbob_09_02
bbob_10_01
bbob_10_02
bbob_11_01
bbob_11_02
bbob_12_01
bbob_12_02
bbob_13_01
bbob_13_02
bbob_14_01
bbob_14_02
bbob_15_01
bbob_15_02
bbob_16_01
bbob_16_02
bbob_17_01
bbob_17_02
bbob_18_01
bbob_18_02
bbob_19_01
bbob_19_02
bbob_20_01
bbob_20_02
bbob_21_01
bbob_21_02
bbob_22_01
bbob_22_02
bbob_23_01
bbob_23_02
bbob_24_01
bbob_24_02

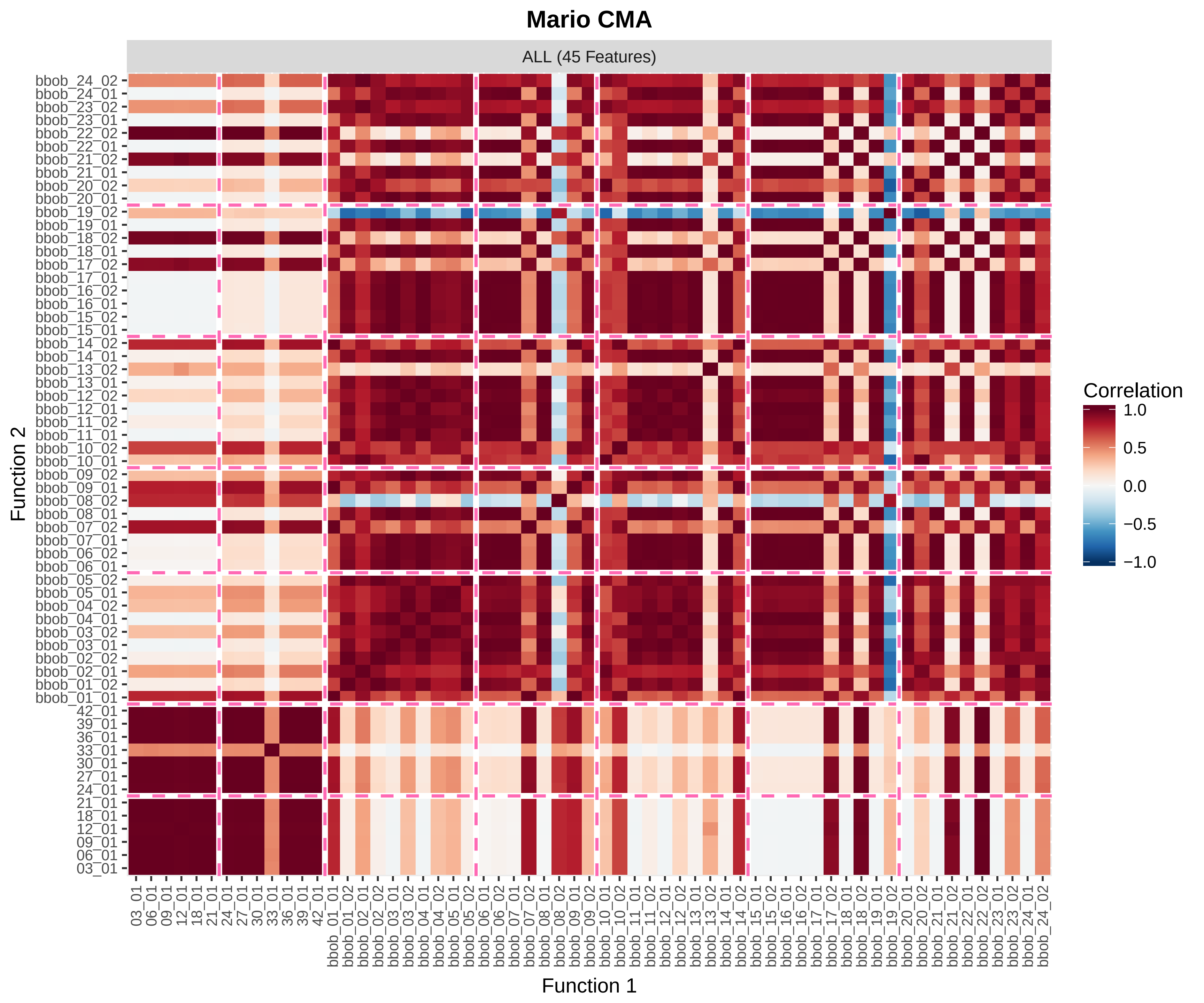


Mario CMA
IC (5 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0



Mario CMA
ELA_META (9 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0


Mario CMA
ELA_DISTR (3 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0
bbob_24_02
bbob_24_01
bbob_23_02
bbob_23_01
bbob_22_02
bbob_22_01
bbob_21_02
bbob_21_01
bbob_20_02
bbob_20_01
bbob_19_02
bbob_19_01
bbob_18_02
bbob_18_01
bbob_17_02
bbob_17_01
bbob_16_02
bbob_16_01
bbob_15_02
bbob_15_01
bbob_14_02
bbob_14_01
bbob_13_02
bbob_13_01
bbob_12_02
bbob_12_01
bbob_11_02
bbob_11_01
bbob_10_02
bbob_10_01
bbob_09_02
bbob_09_01
bbob_08_02
bbob_08_01
bbob_07_02
bbob_07_01
bbob_06_02
bbob_06_01
bbob_05_02
bbob_05_01
bbob_04_02
bbob_04_01
bbob_03_02
bbob_03_01
bbob_02_02
bbob_02_01
bbob_01_02
bbob_01_01
42_01
39_01
36_01
33_01
30_01
27_01
24_01
21_01
18_01
12_01
09_01
06_01
03_01


Mario CMA
ALL (45 Features)
Function 2
Function 1
Correlation
1.0
0.5
0.0
−0.5
−1.0