Learning music production practice through evolutionary algorithms

Alex Wilson / Bruno Fazenda

Acoustics Research Centre,
University of Salford,
UK

Intelligent Music Production

- The field of intelligent music production has been an active research topic for over a decade.

- The aim is to develop systems which are capable of performing common tasks in music production, such as level-balancing, equalisation, panning, dynamic range compression and application of artificial reverberation.

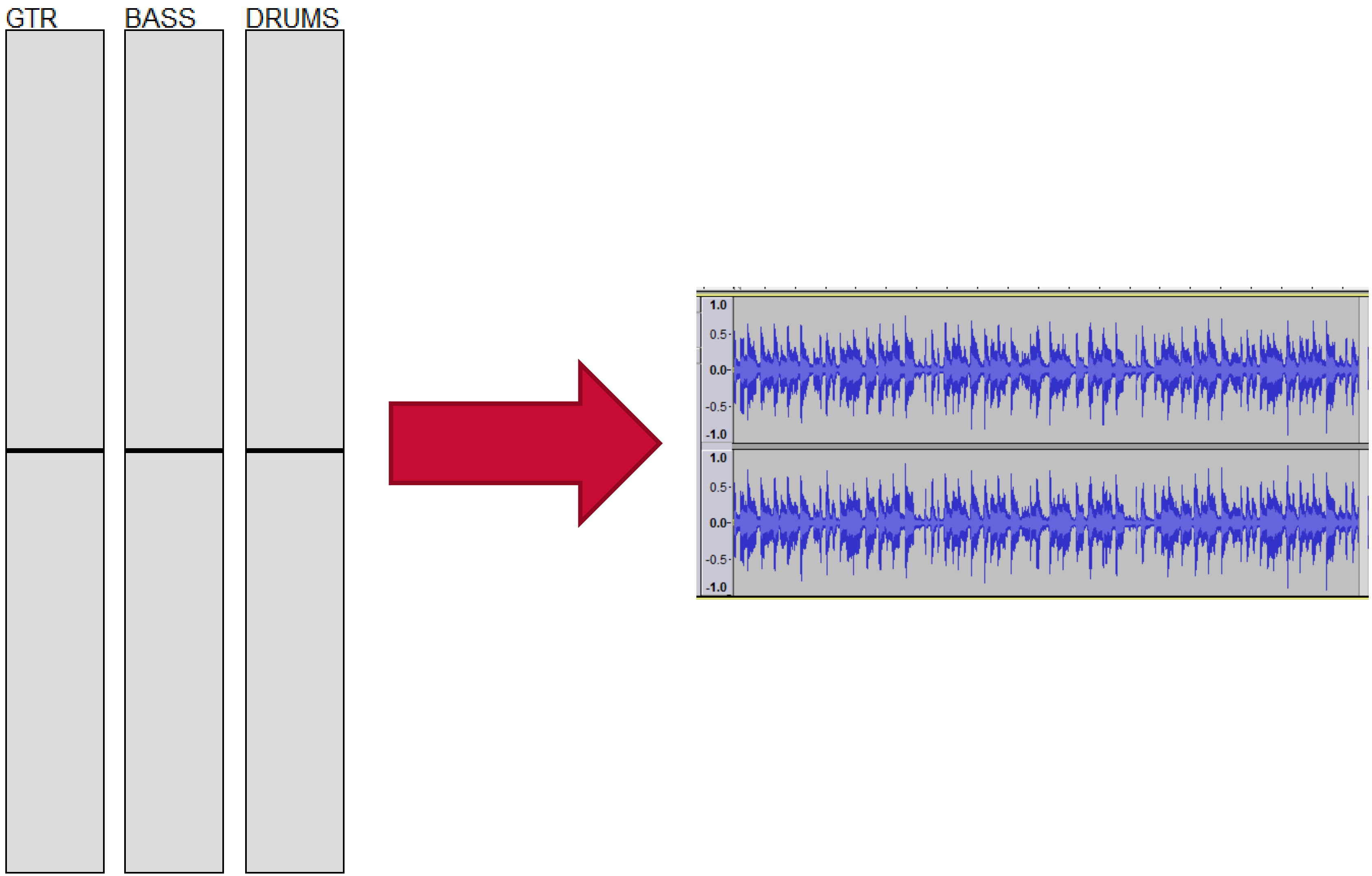
Intelligent Music Production

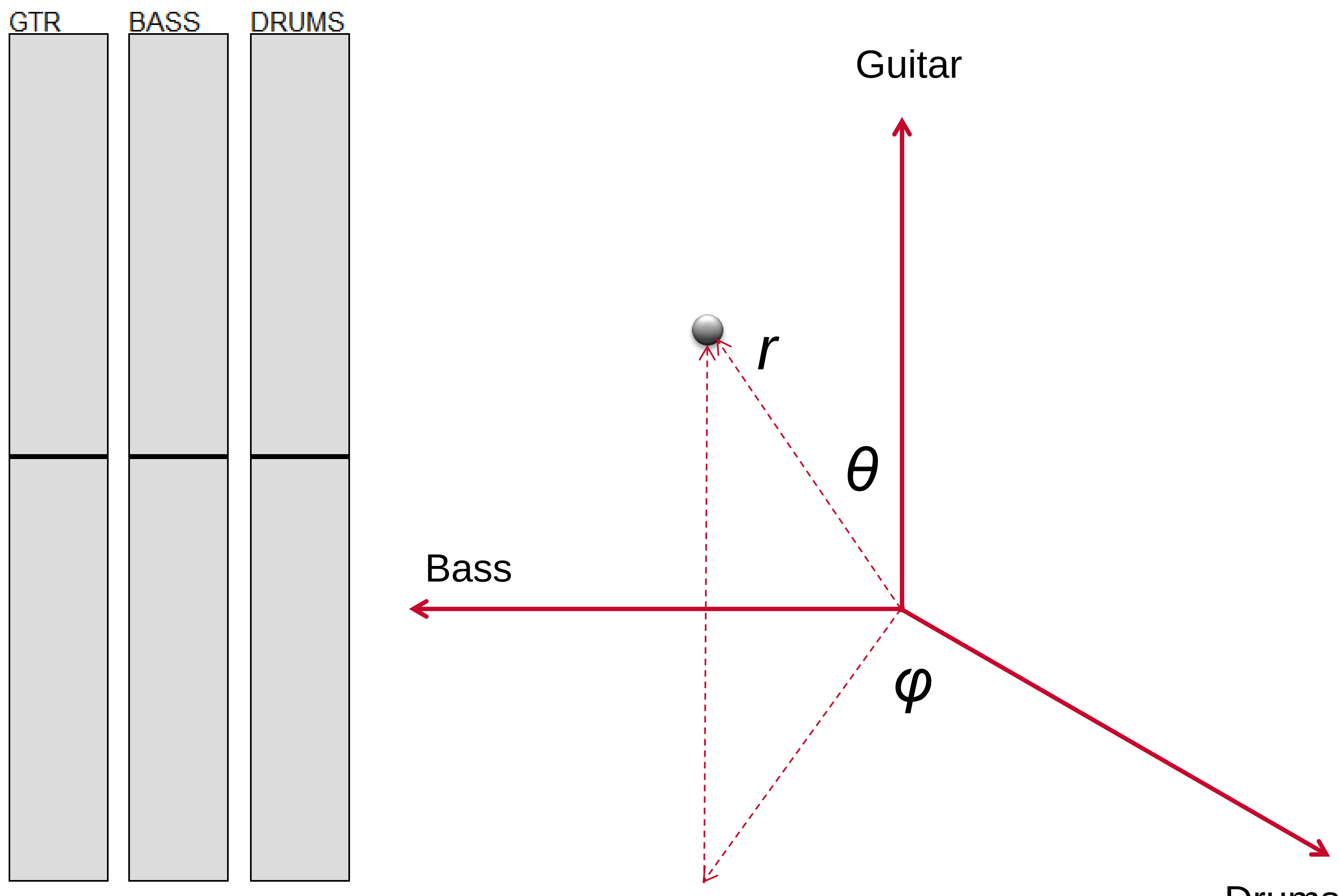
Typical system aesthetics and constraints

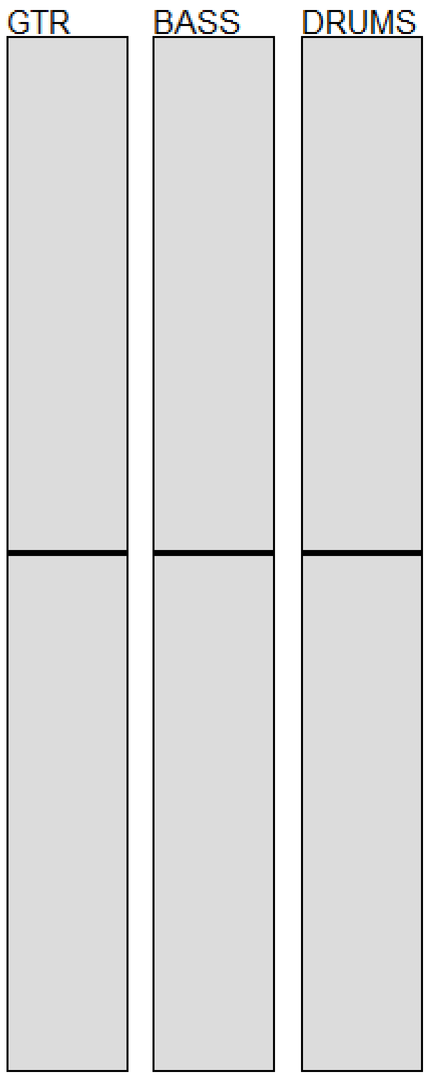
- Real-time (designed for “live” environment)
- Applies audio-specific processing, based on “best-practice”
- Searches for the “best” mix, **objectively**, <u>irrespective of user</u>

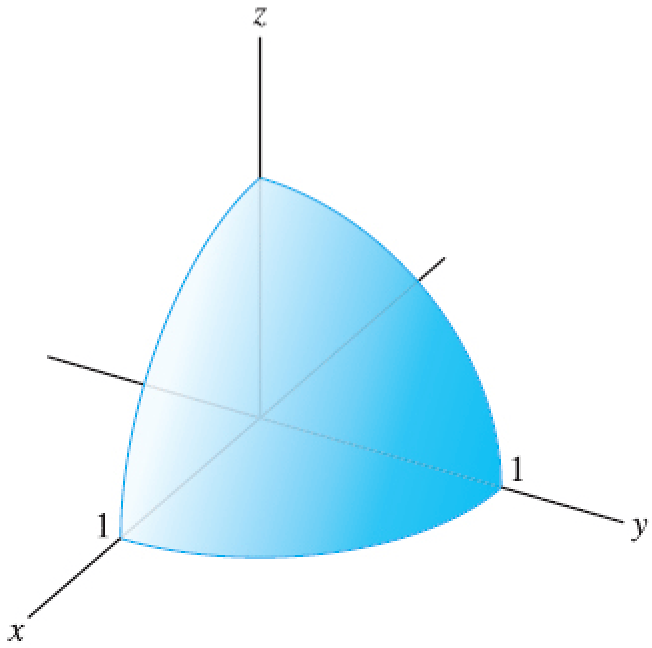
Proposed system aesthetics and constraints

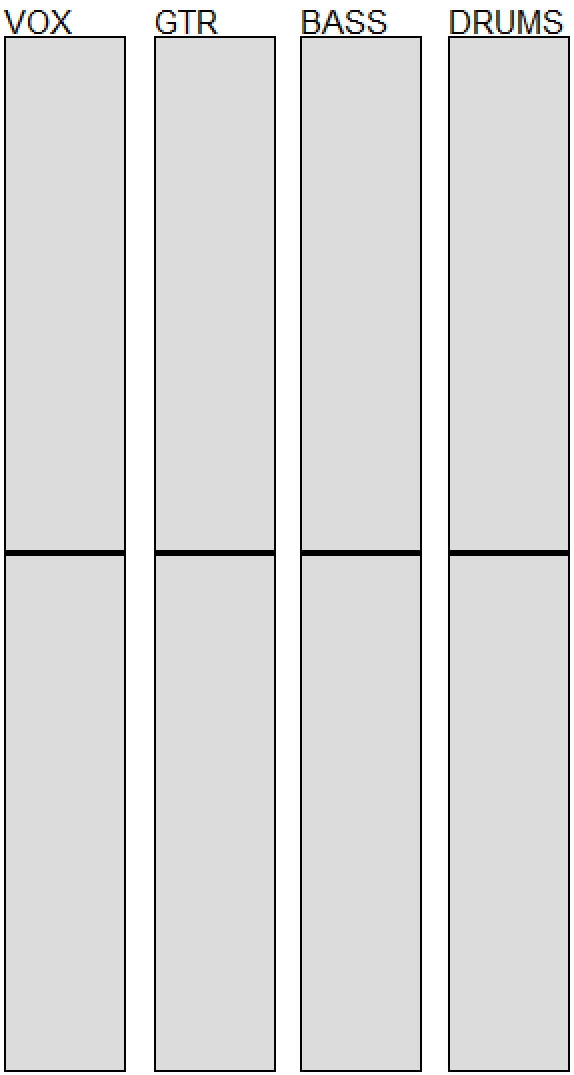
- Off-line (designed for “studio” environment)
- Makes no assumptions about “best-practice”
- Searches for the “best” mix, **subjectively**, <u>according to the user</u>

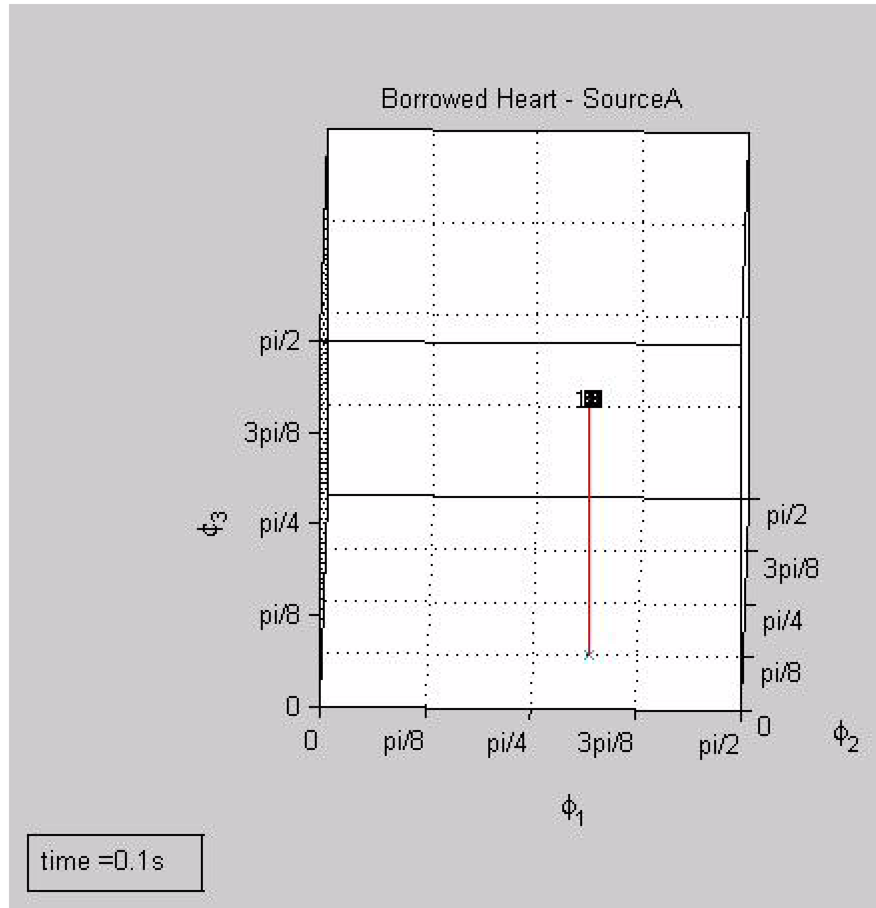

GTR
BASS
DRUMS
1.0
0.5
0.0
-0.5
-1.0
1.0
0.5
0.0
-0.5
-1.0


GTR
BASS
DRUMS
Guitar
r
θ
Bass
φ
Drums

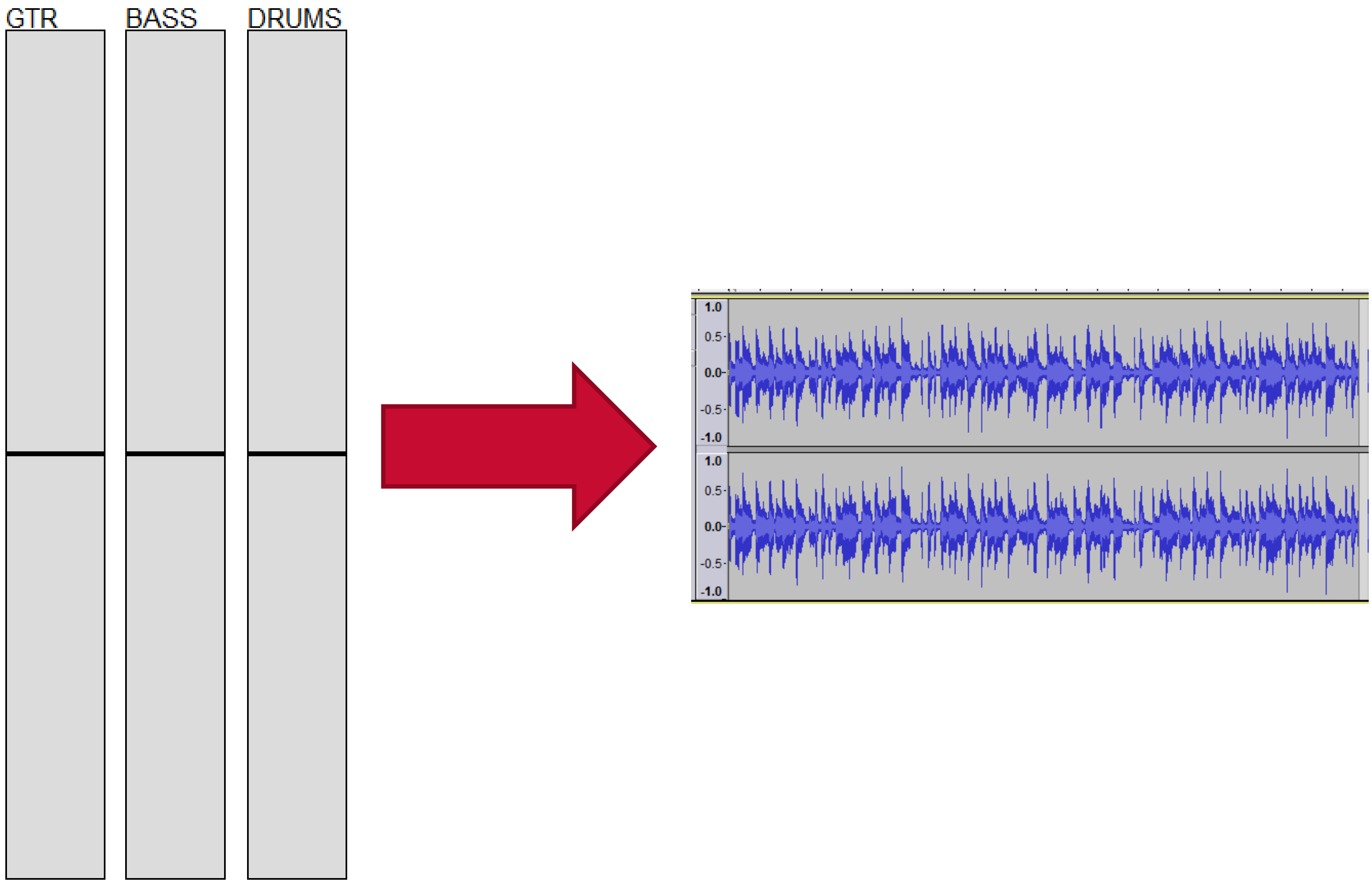

GTR
BASS
DRUMS

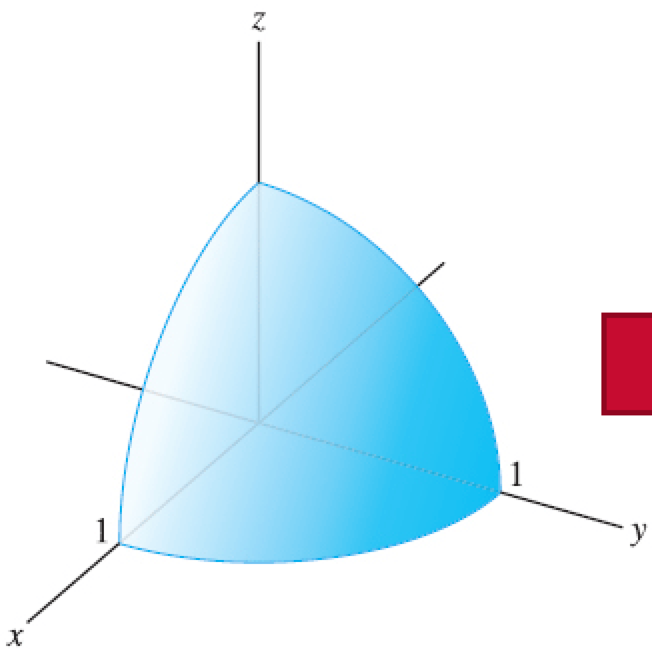

z
1
1
x
y

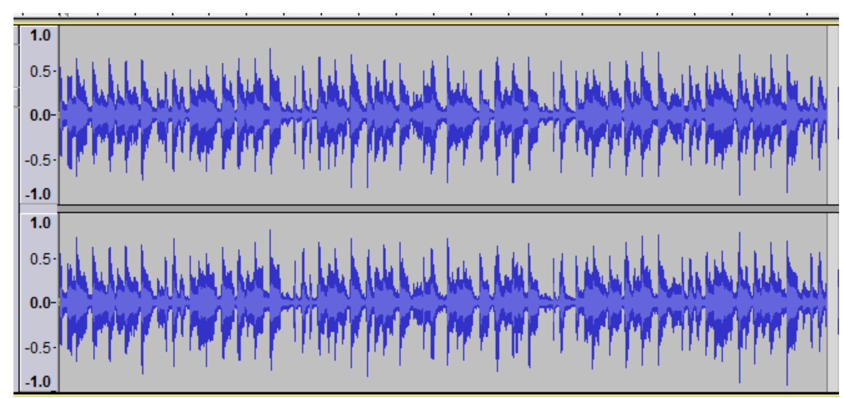




"Navigating the mix-space: theoretical and practical level-balancing technique in multitrack music mixtures", **Wilson/Fazenda, 2015**

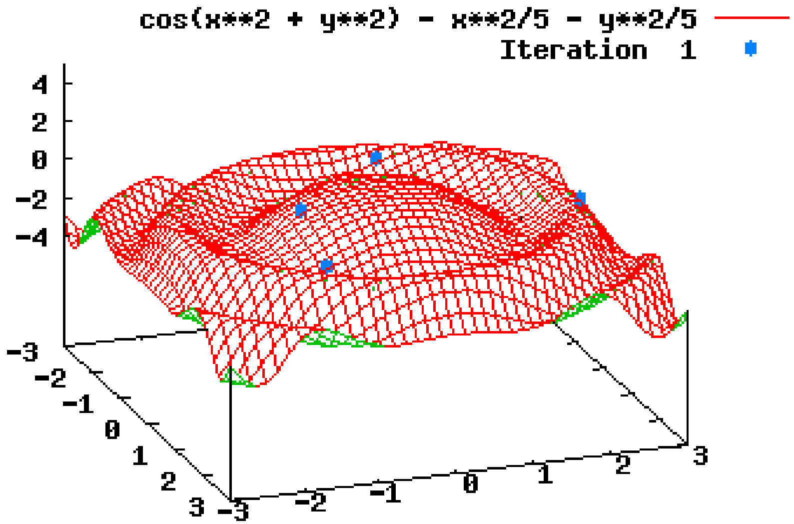

GTR
BASS
DRUMS
1.0
0.5
0.0
-0.5
-1.0
1.0
0.5
0.0
-0.5
-1.0


z
y
x
1
1


1.0
0.5
0.0
-0.5
-1.0
1.0
0.5
0.0
-0.5
-1.0

Evolutionary Algorithms

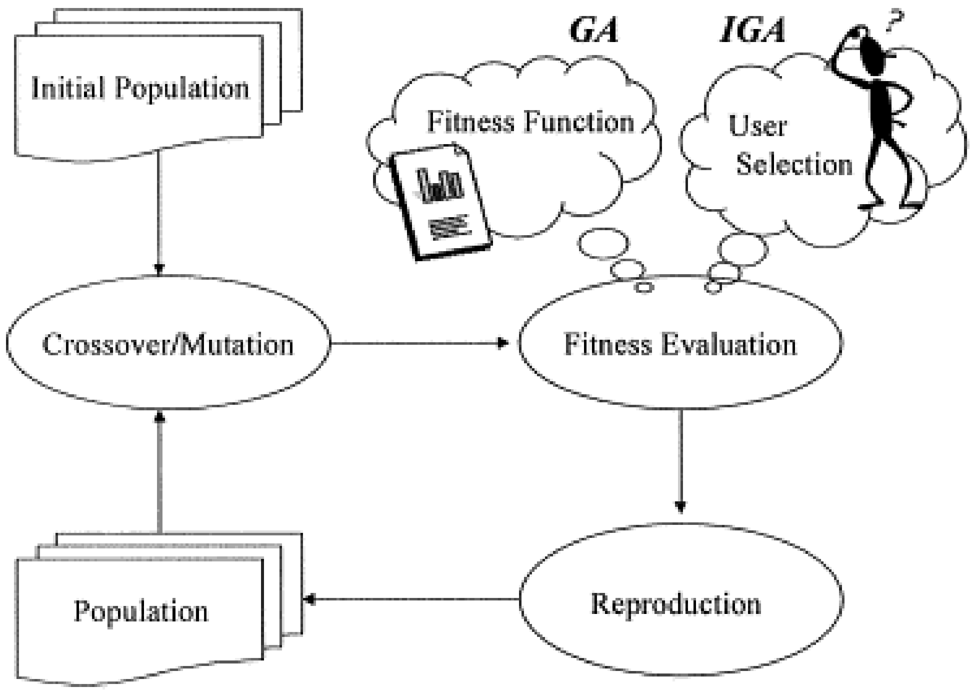
- Generic population-based metaheuristic optimisation algorithm
- Can be used for global optimisation
- Techniques include:
  - **Genetic algorithm**
  - Particle Swarm Optimisation
  - Ant Colony Optimisation
  - Bees Algorithm



http://ittner.github.io/abelhas/

(Interactive) Evolutionary Algorithms

- Particularly suitable to aesthetic design problems
  - Non-linear, not deterministic



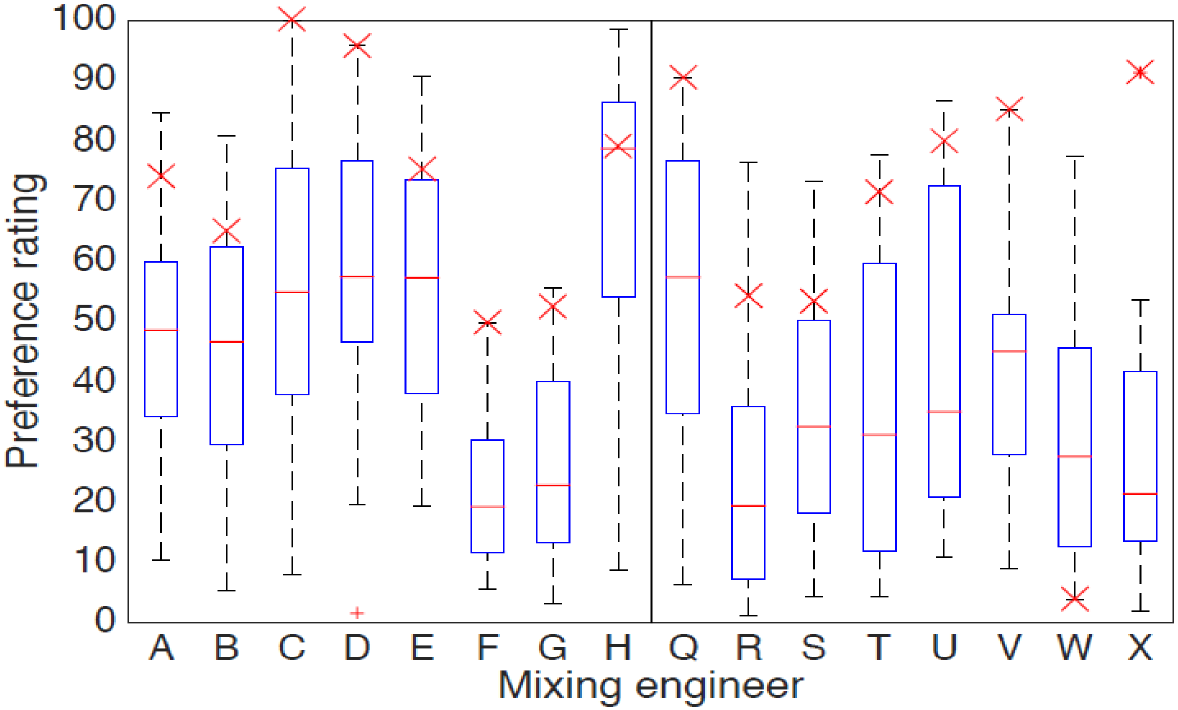
(Interactive) Evolutionary Algorithms

- Particularly suitable to aesthetic design problems
  - Non-linear, not deterministic

- IEA has applied to…
  - Fashion design (Kim/Cho, 2000)
  - Structural design (O'Neill et al. 2010)
  - Logo design (O'Neill/Brabazon, 2008)

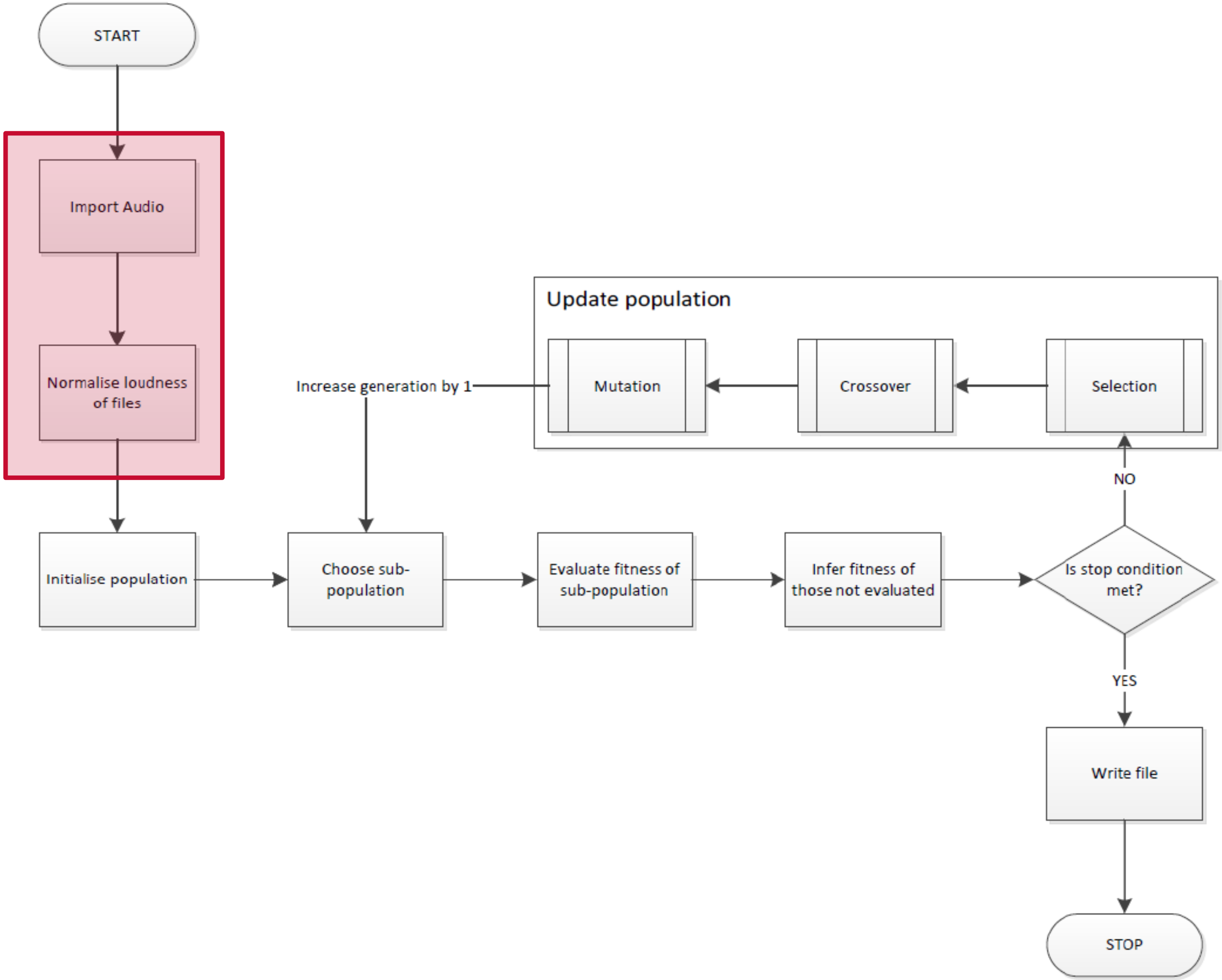
(Interactive) Evolutionary Algorithms

- Why have user-assisted system?
  - The *"best"* mix won't be the best for everyone
  - The user can help guide the mixing system towards their ideal mix



"Perceptual Evaluation of Music Mixing Practices", **De Man et al., 2015**

Flowchart



START
Import Audio
Normalise loudness of files
Initialise population
Choose sub-population
Evaluate fitness of sub-population
Infer fitness of those not evaluated
Is stop condition met?
NO
YES
Update population
Selection
Crossover
Mutation
Increase generation by 1
Write file
STOP

Import and normalise

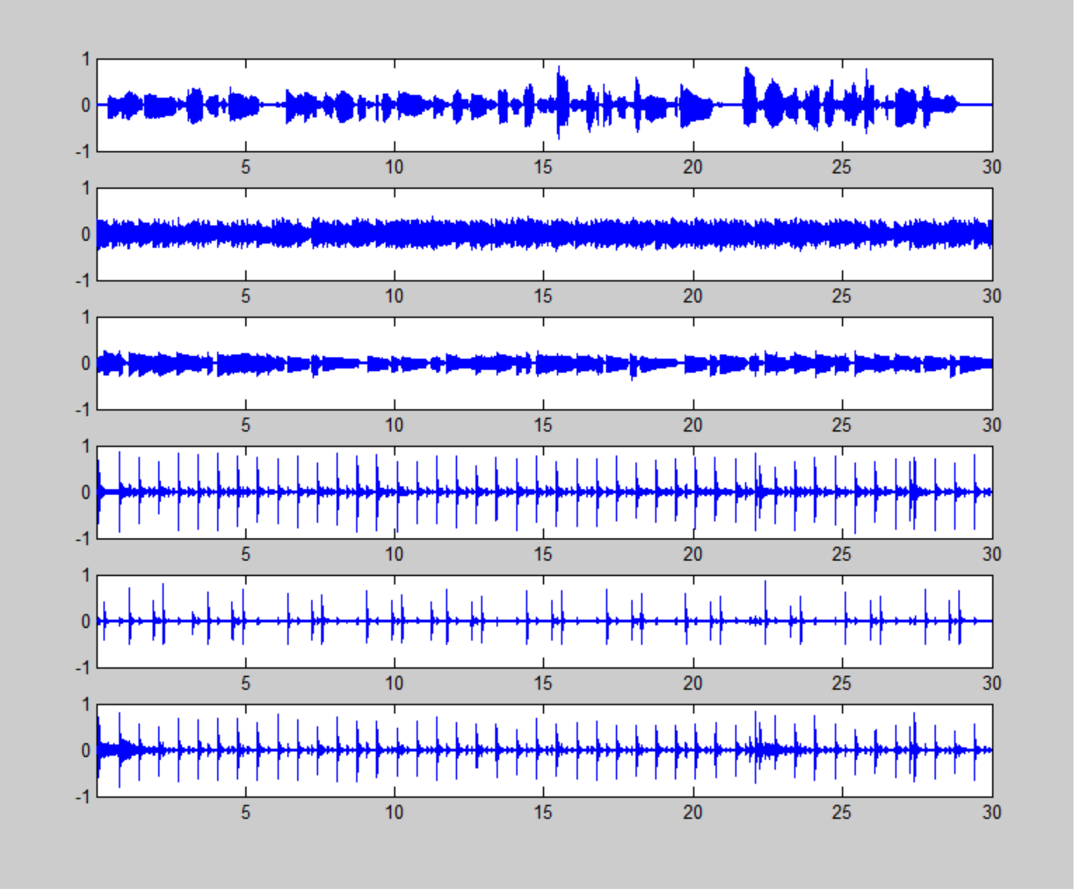
Loudness normalisation

- Using modified version of BS.1770, to better account for narrowband signals

For this demo, 6 tracks are used

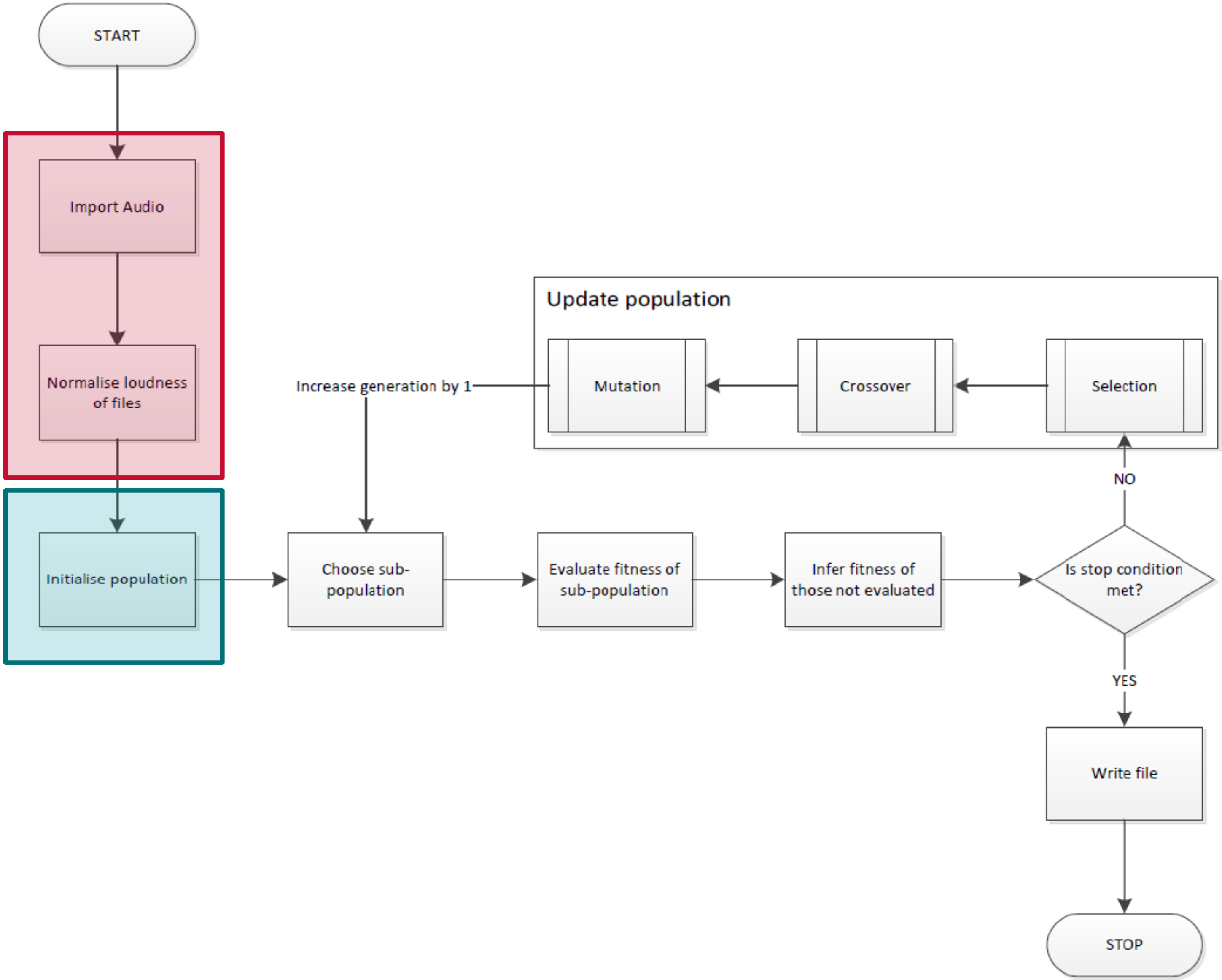
- Vocals
- Guitar
- Bass
- Snare drum
- Kick drum
- Drum overhead

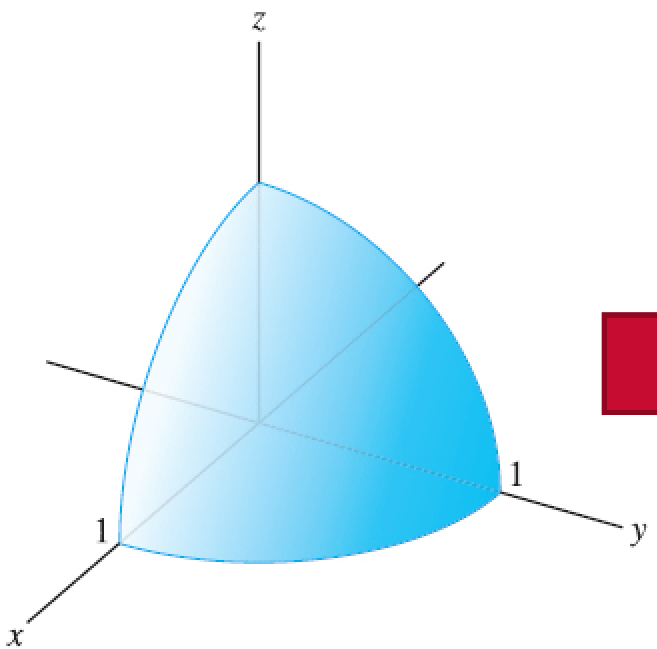
No EQ, Panning, Compression, Reverb, …, etc.

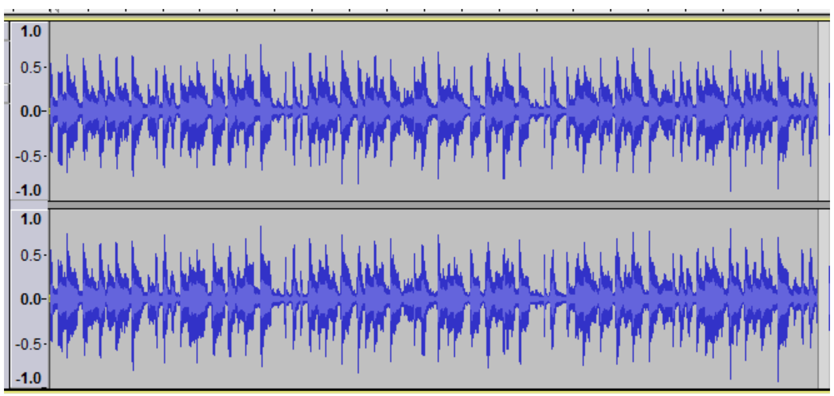


"Loudness measurement of multitrack audio content using modifications of ITU-R BS. 1770", **Pestana et. al., 2013**

Flowchart


START
Import Audio
Normalise loudness of files
Initialise population
Choose sub-population
Evaluate fitness of sub-population
Infer fitness of those not evaluated
Is stop condition met?
NO
YES
Update population
Selection
Crossover
Mutation
Increase generation by 1
Write file
STOP


z
1
1
y
x

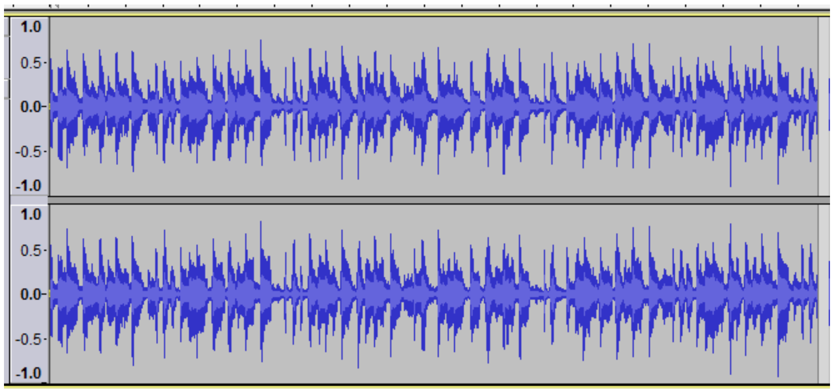

1.0
0.5
0.0
-0.5
-1.0
1.0
0.5
0.0
-0.5
-1.0

6-tracks, therefore...

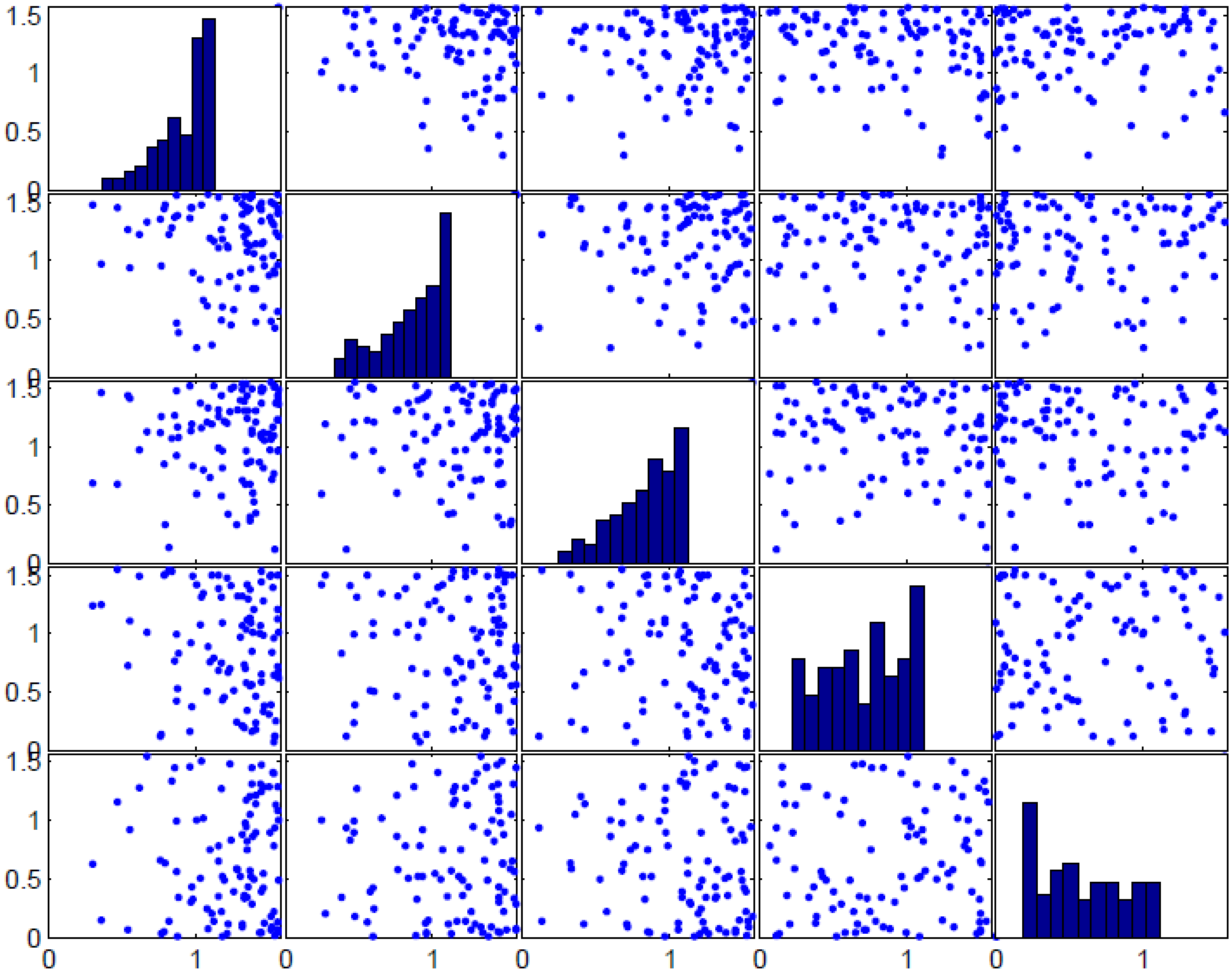
...mixes are on the surface of a sphere in 6-dimensional space...

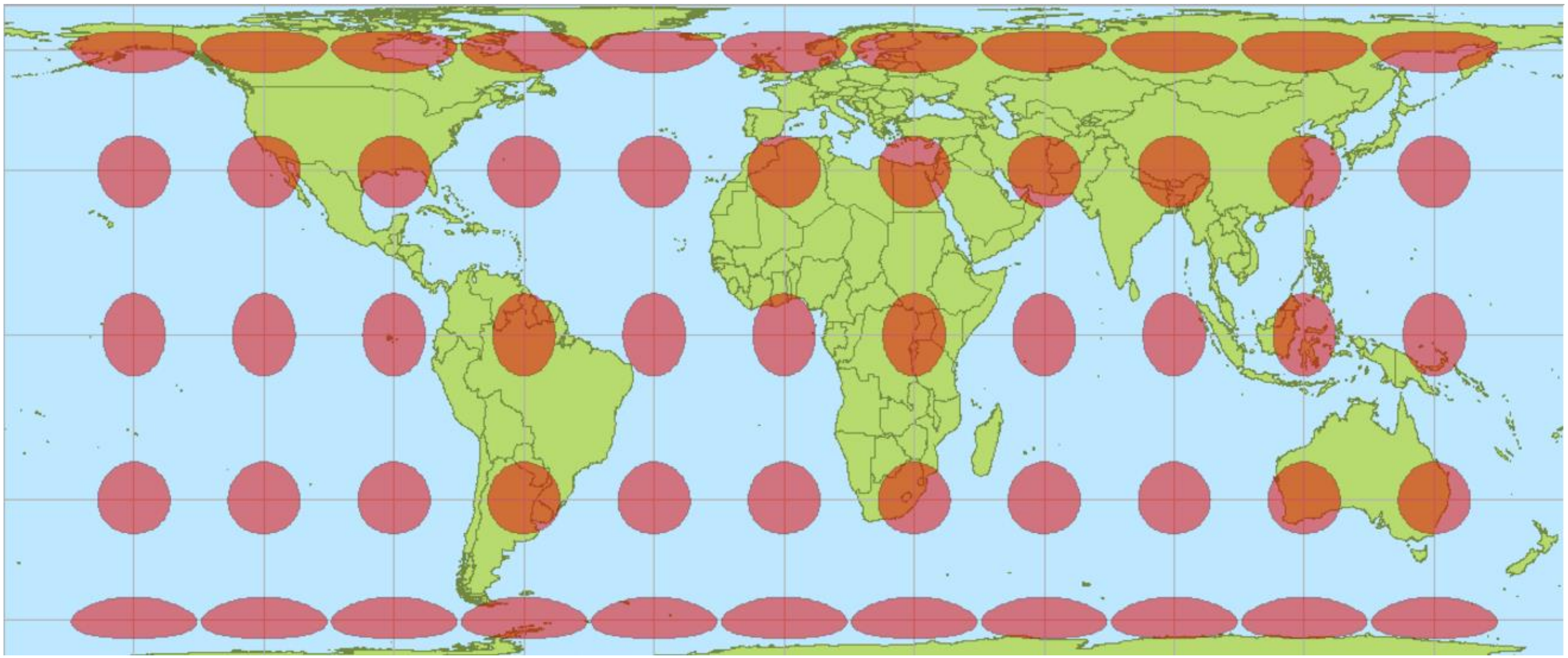
...known as a 5-sphere, as it’s surface is 5D.

φ_1
φ_2
φ_3
φ_4
φ_5

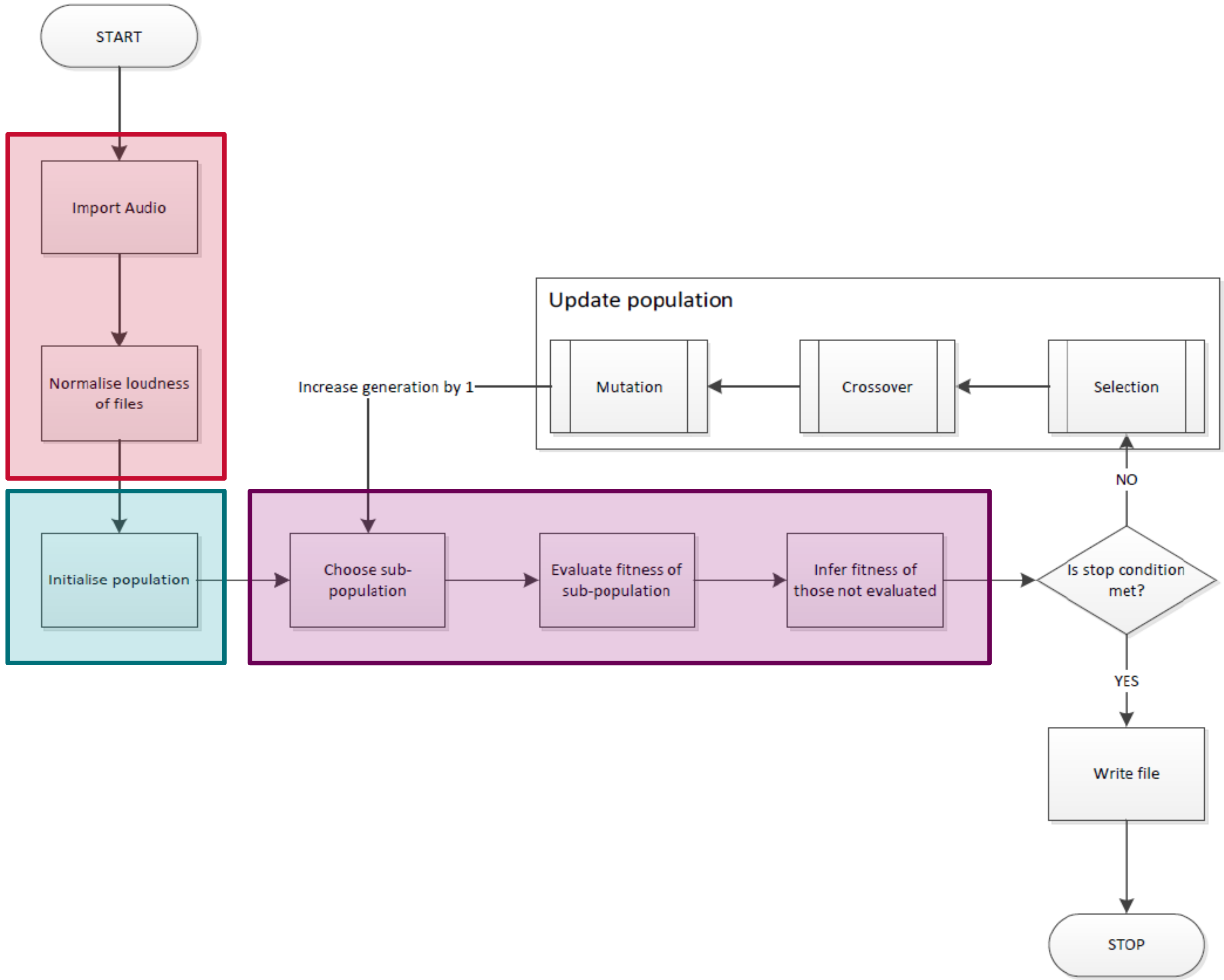


Initial population





Flowchart



START
Import Audio
Normalise loudness of files
Initialise population
Choose sub-population
Evaluate fitness of sub-population
Infer fitness of those not evaluated
Is stop condition met?
NO
YES
Update population
Mutation
Crossover
Selection
Increase generation by 1
Write file
STOP

Clustering

- Initialise()
- While not end condition do
  - Select()
  - Crossover()
  - Mutate()
  - Do clustering()
  - Pick representatives()
  - Evaluate representatives()
  - Allocate fitness()
- End while

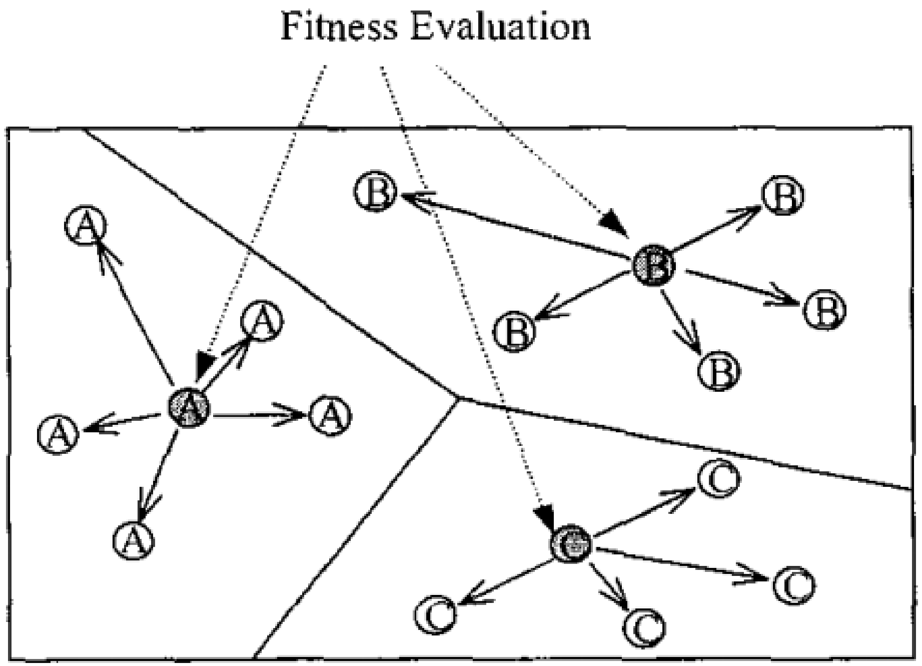
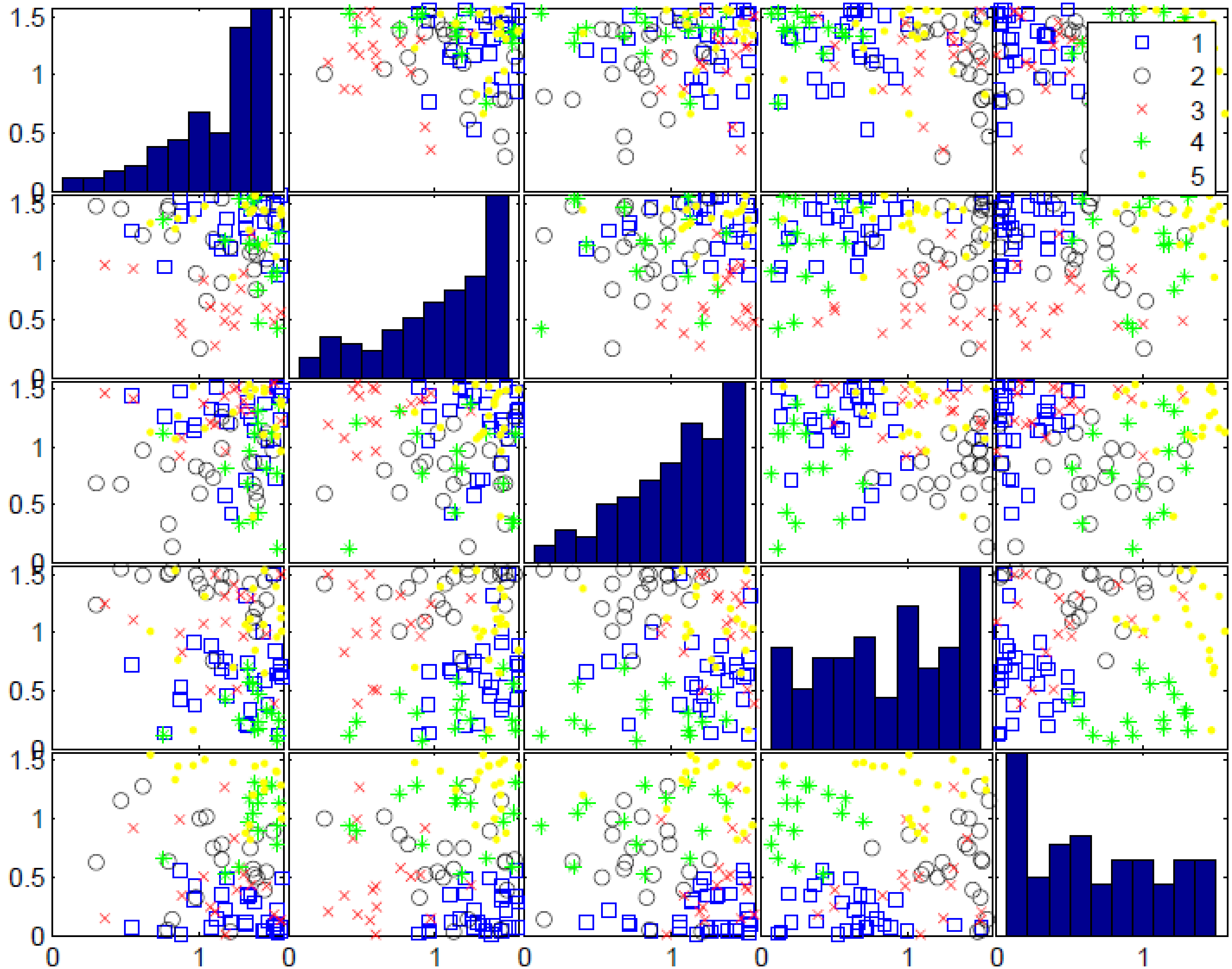


Fig. 6 Sparse fitness evaluation.

"Sparse fitness evaluation for reducing user burden in interactive genetic algorithm", **Lee/Cho, 1999**

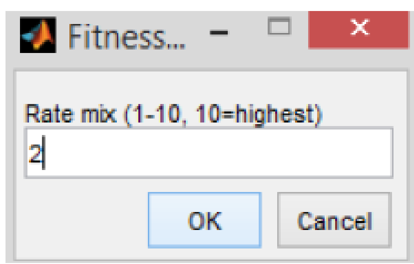
Clustering

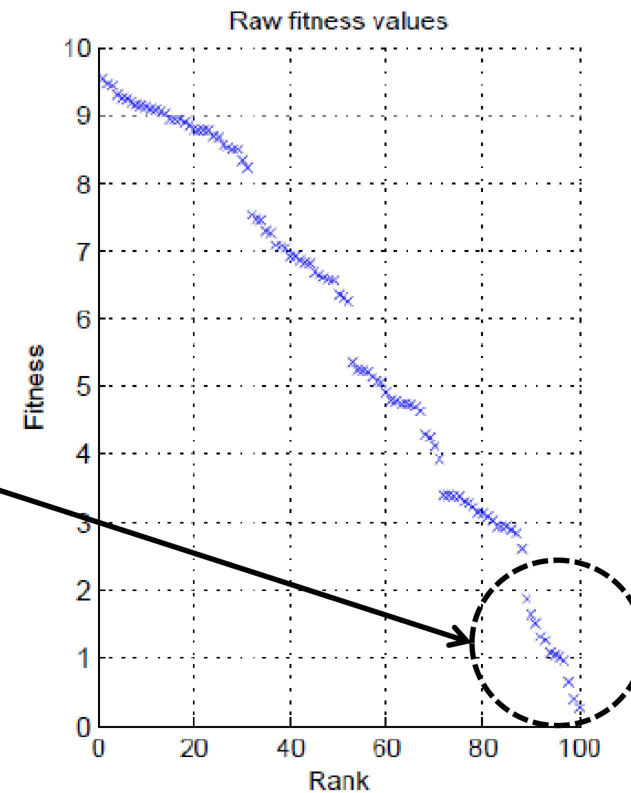


Evaluate fitness

- Get fitness of representatives
- Assign fitness to individuals not evaluated

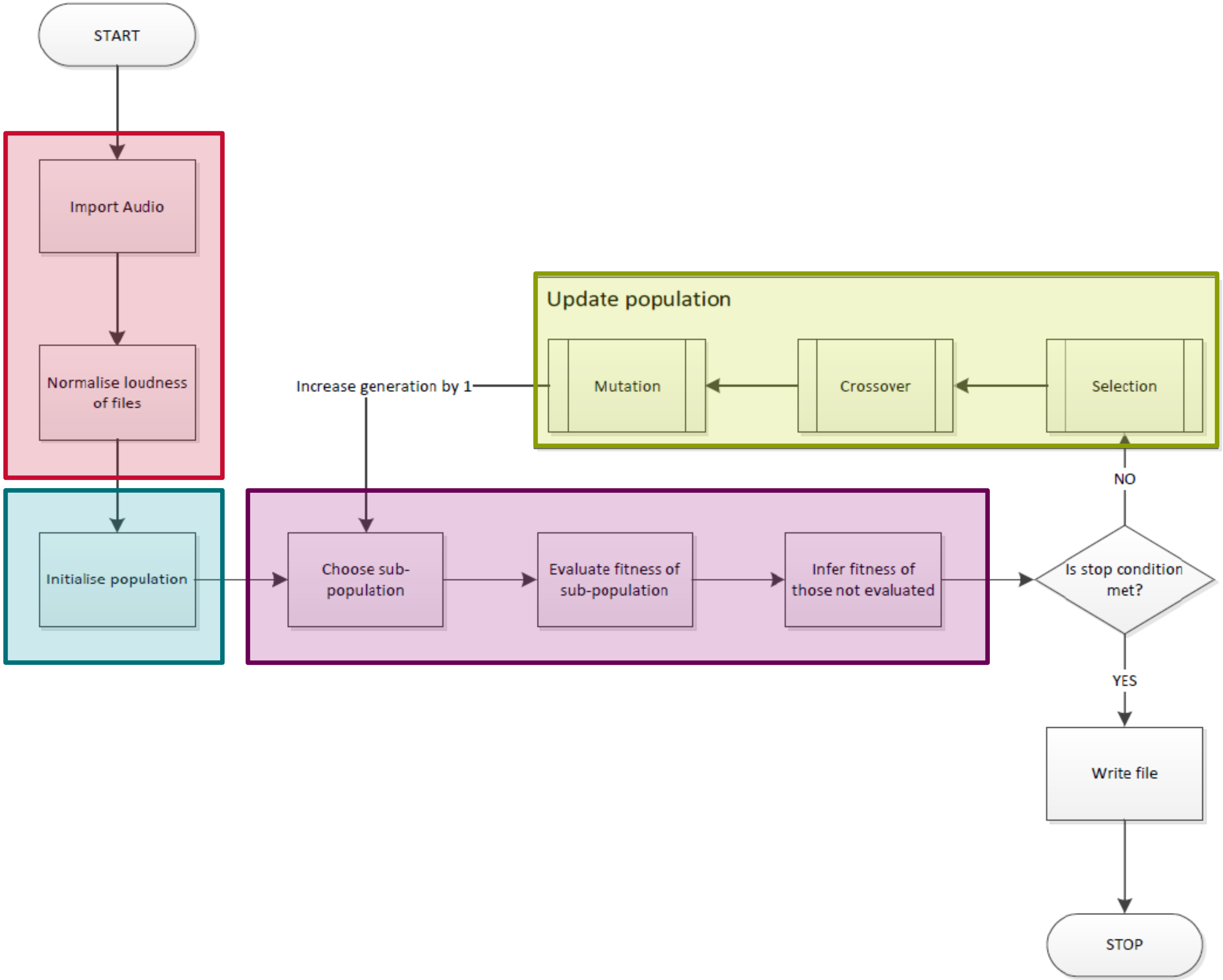
- Subjective fitness
  - “How much do you like this mix?”





  - GA will then maximise this over time

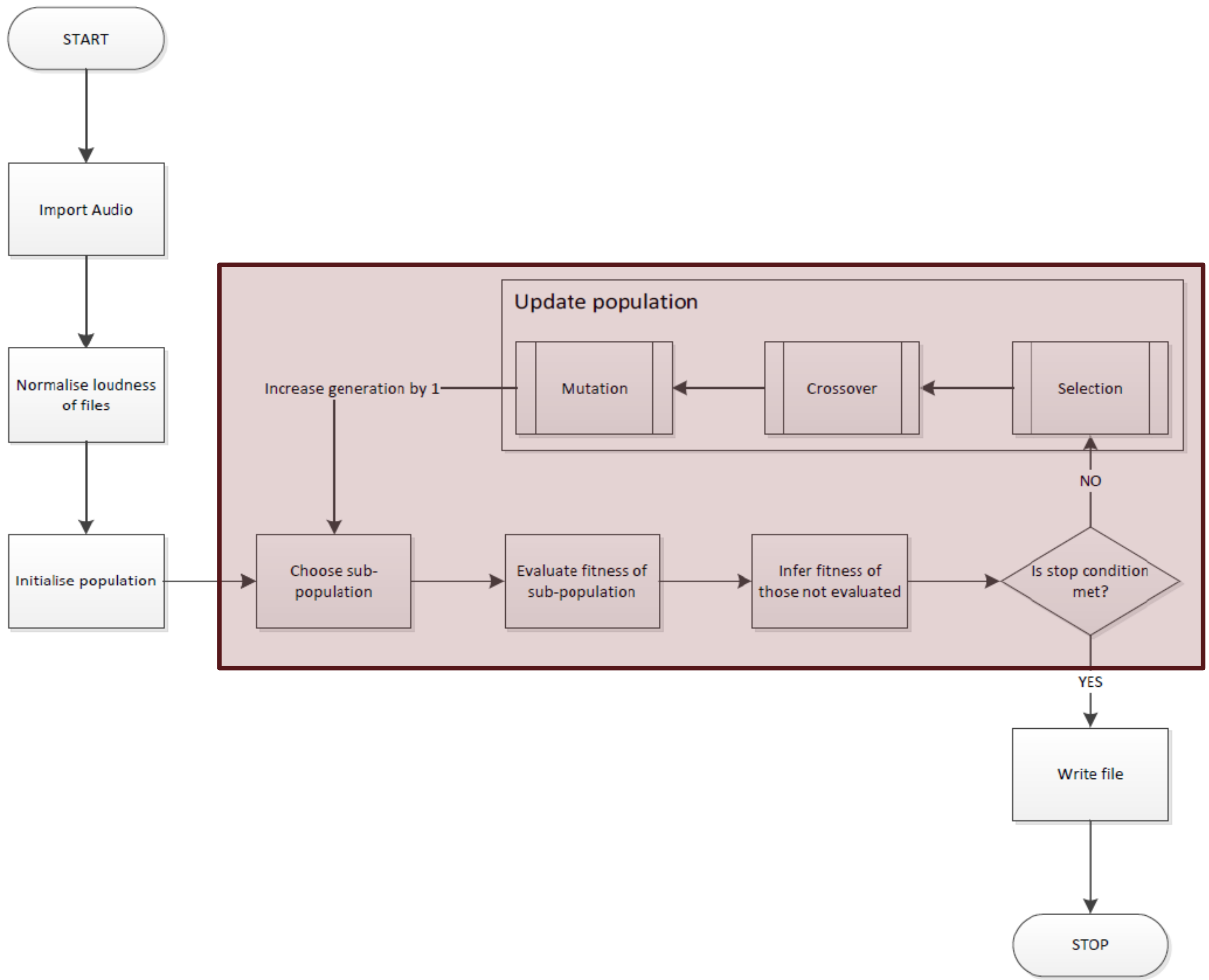
Flowchart



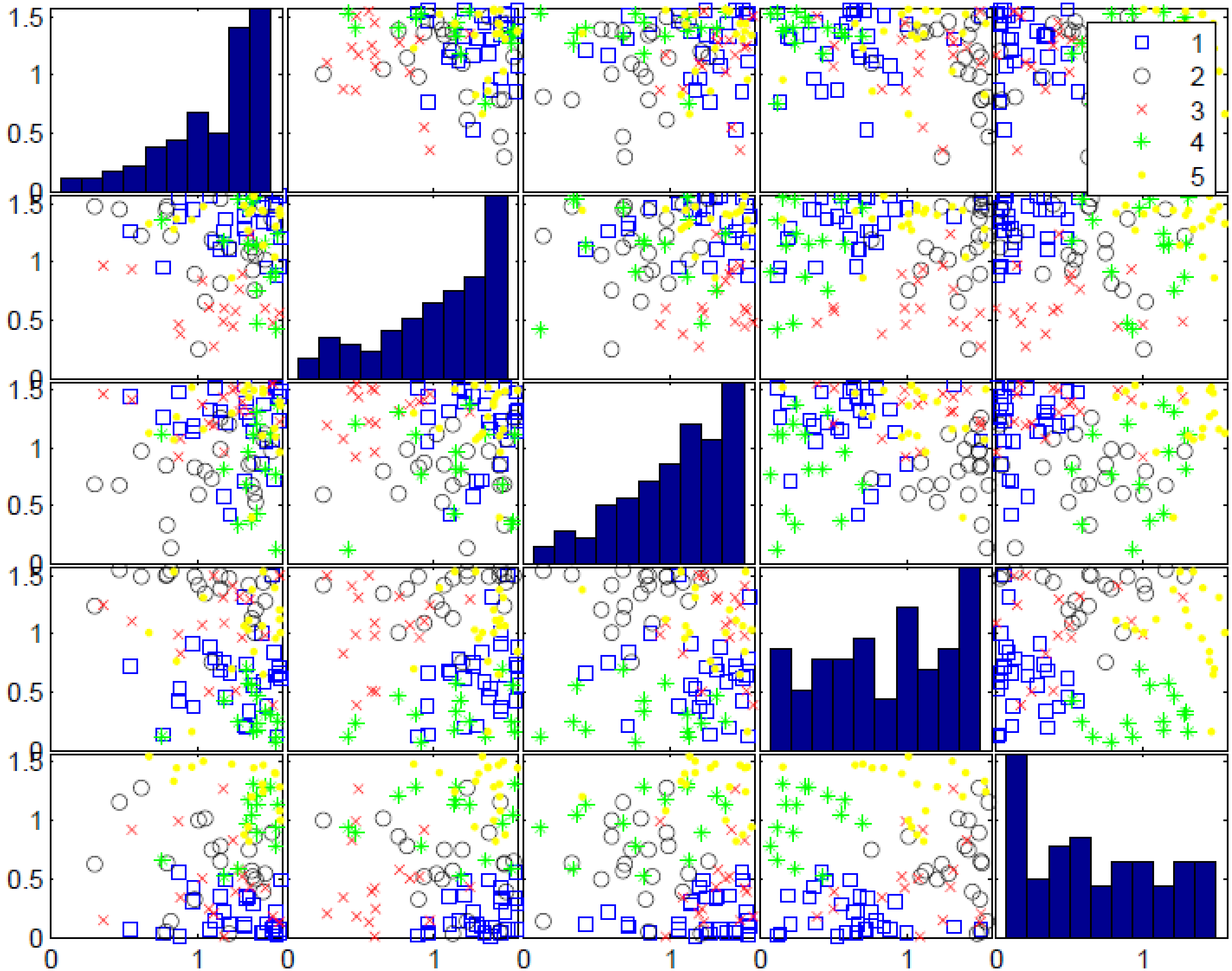
START
Import Audio
Normalise loudness of files
Initialise population
Choose sub-population
Evaluate fitness of sub-population
Infer fitness of those not evaluated
Is stop condition met?
NO
YES
Update population
Mutation
Crossover
Selection
Increase generation by 1
Write file
STOP

Update population

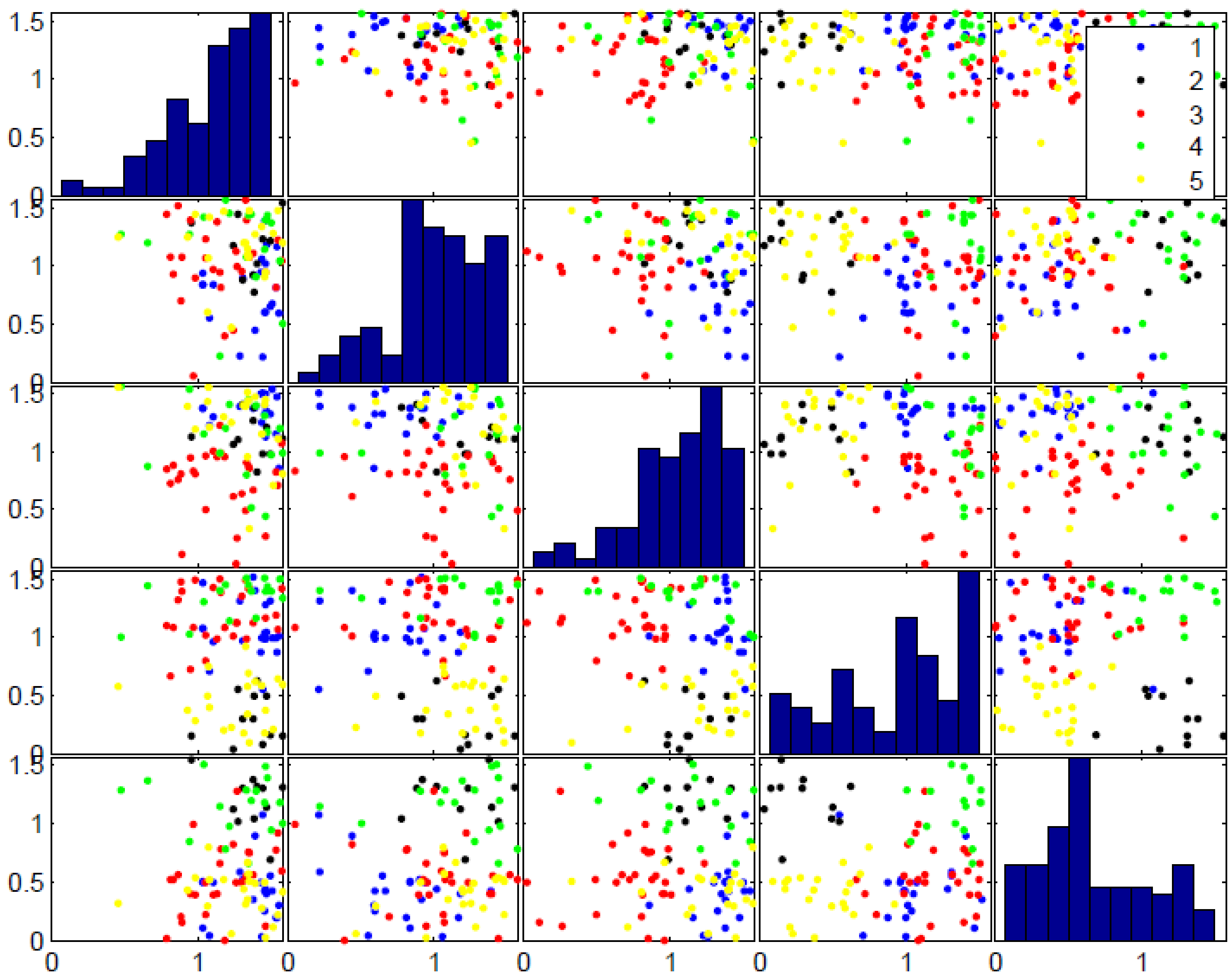
- Fitness Scaling
  - Ranking
- Selection
  - Roulette
- Crossover
  - One-point, **Uniform**, …
- Mutation
  - Random bit flip


START
Import Audio
Normalise loudness of files
Initialise population
Update population
Increase generation by 1
Mutation
Crossover
Selection
Choose sub-population
Evaluate fitness of sub-population
Infer fitness of those not evaluated
Is stop condition met?
NO
YES
Write file
STOP

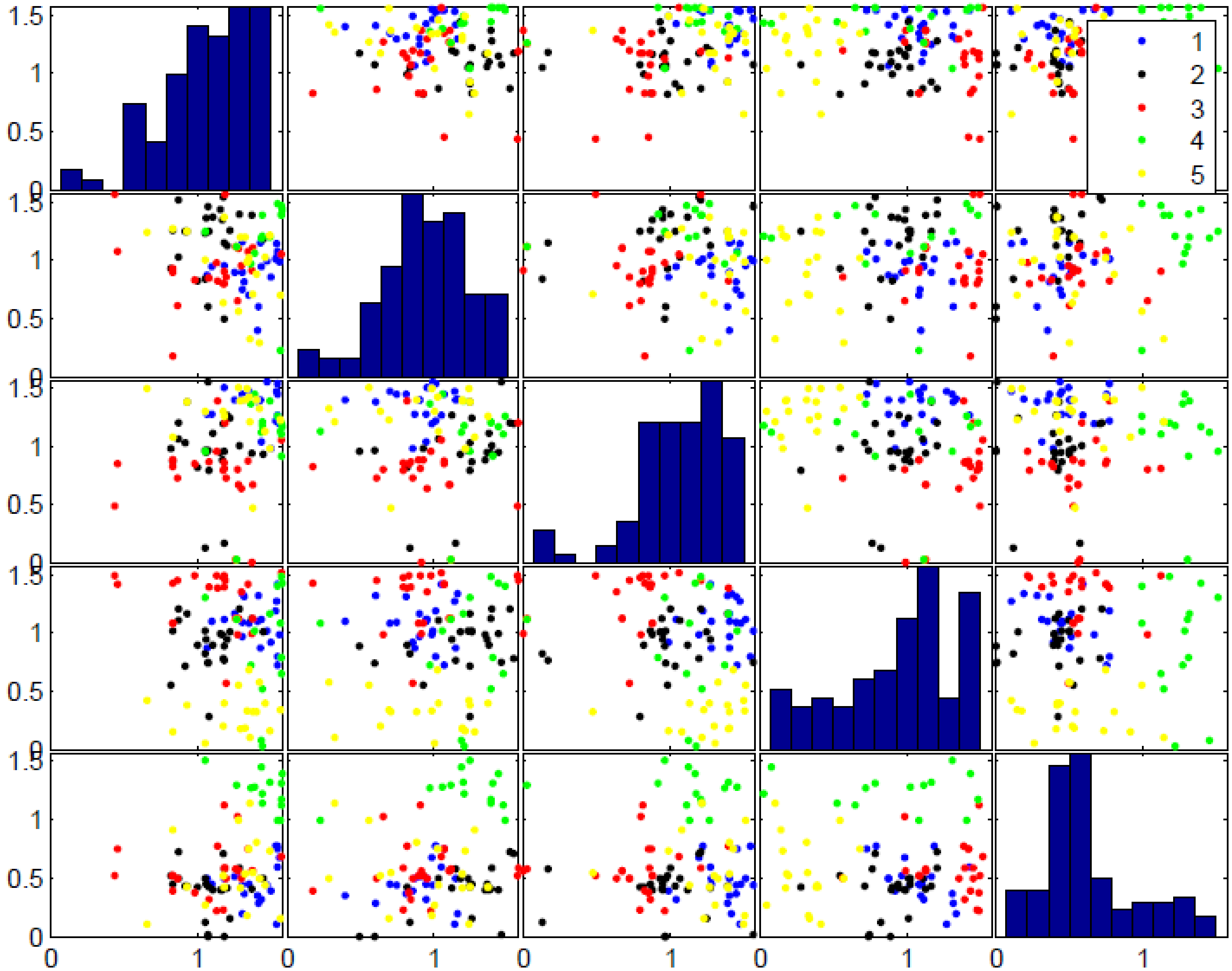
Initial population



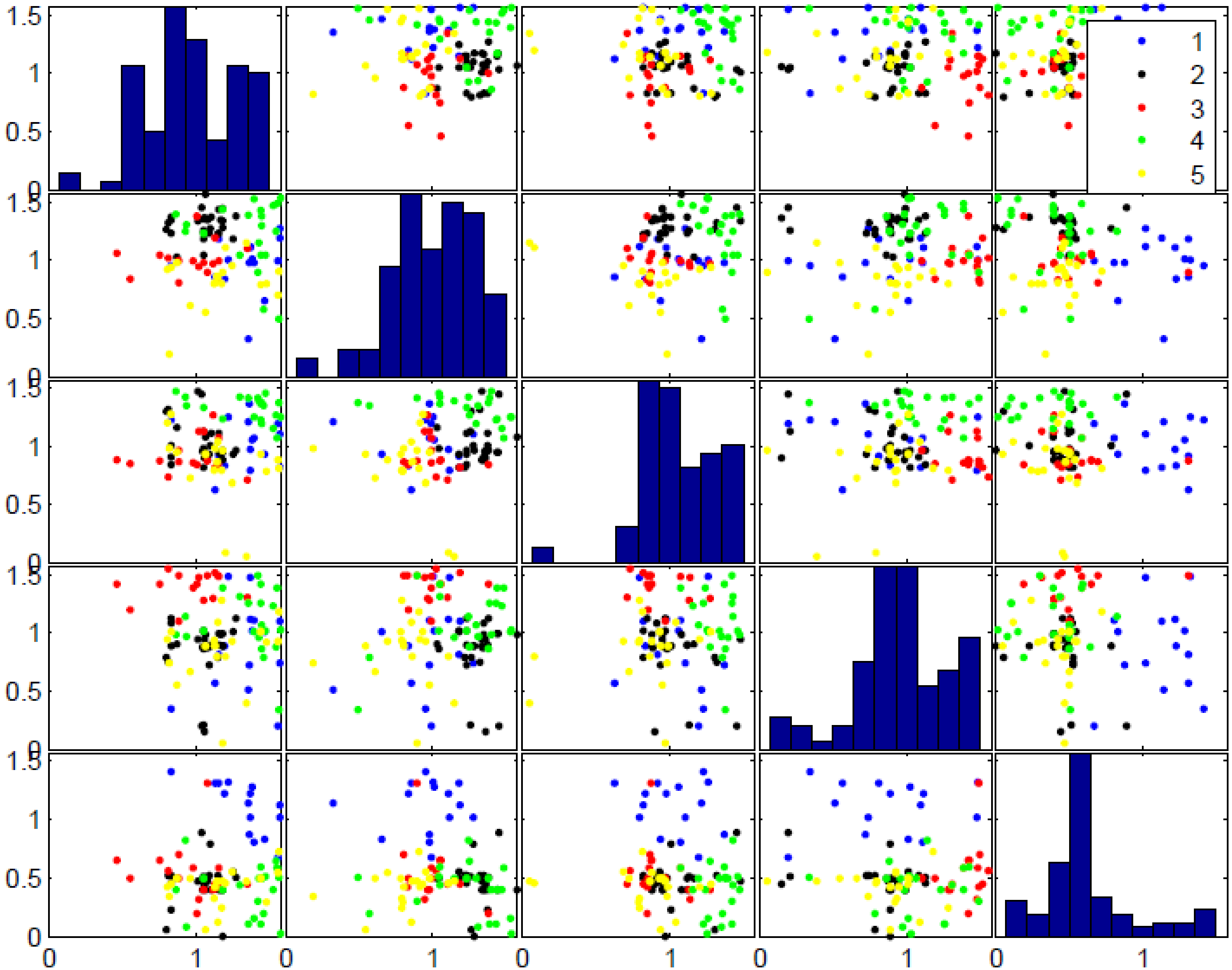
Gen = 1



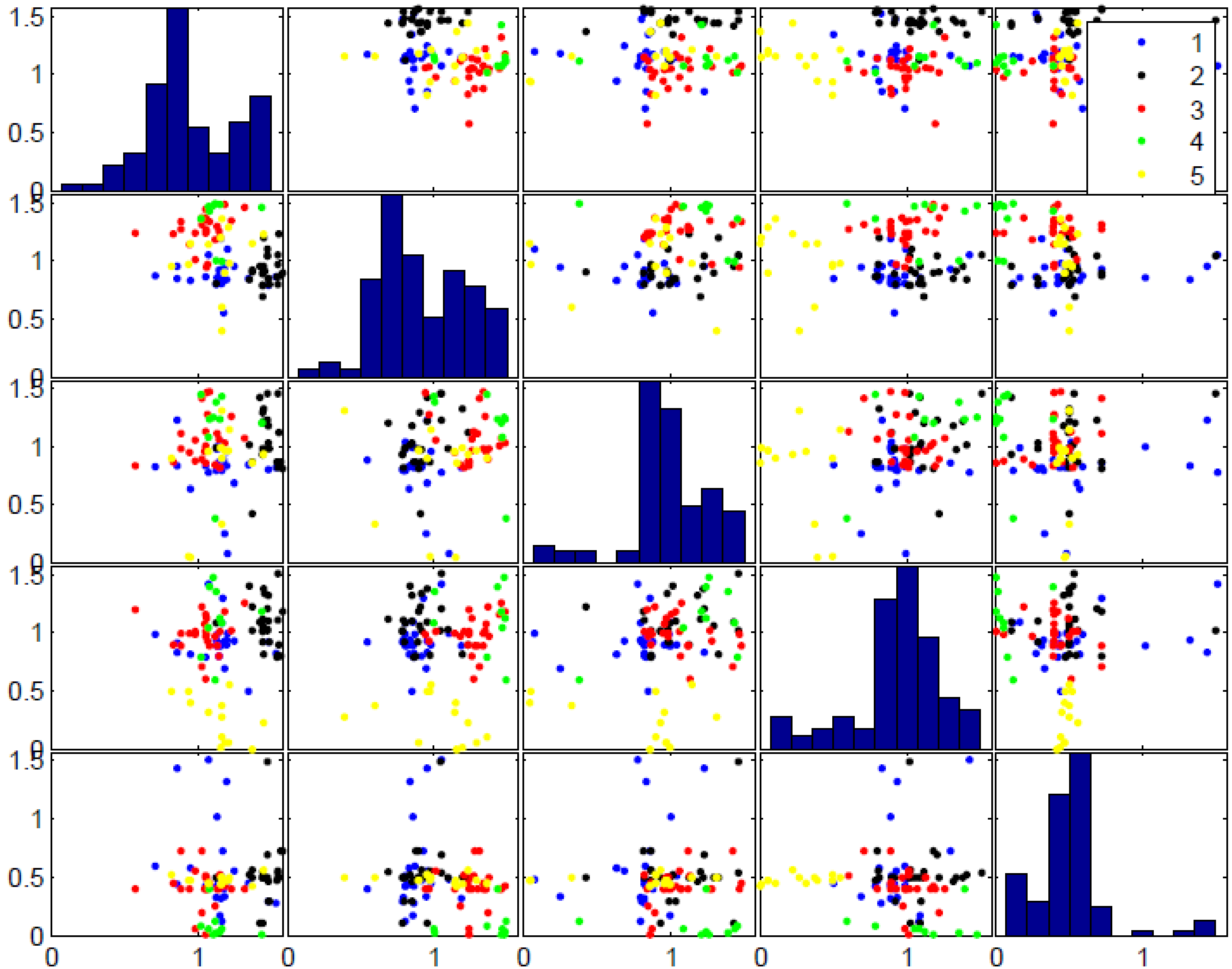
Gen = 2



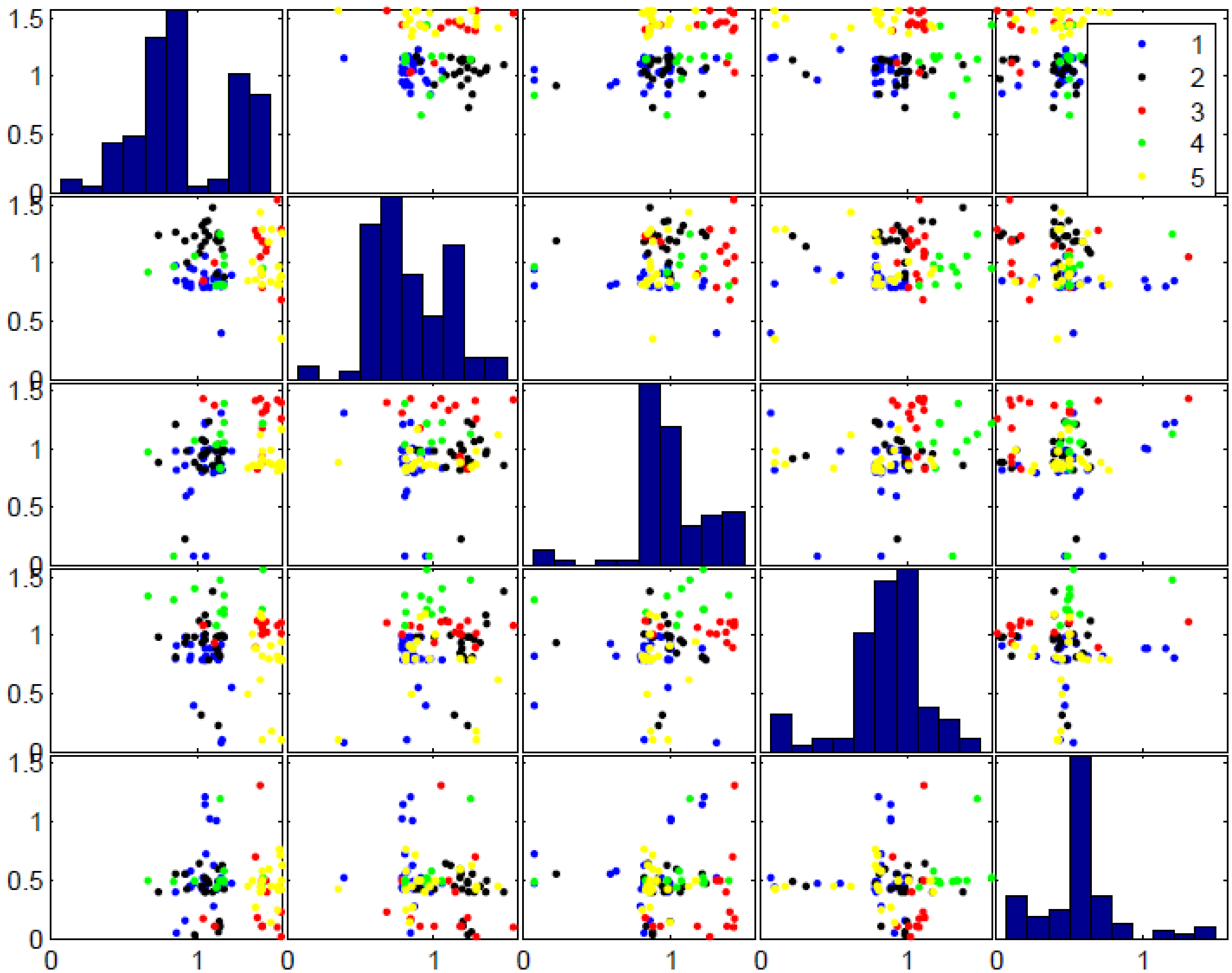
Gen = 3



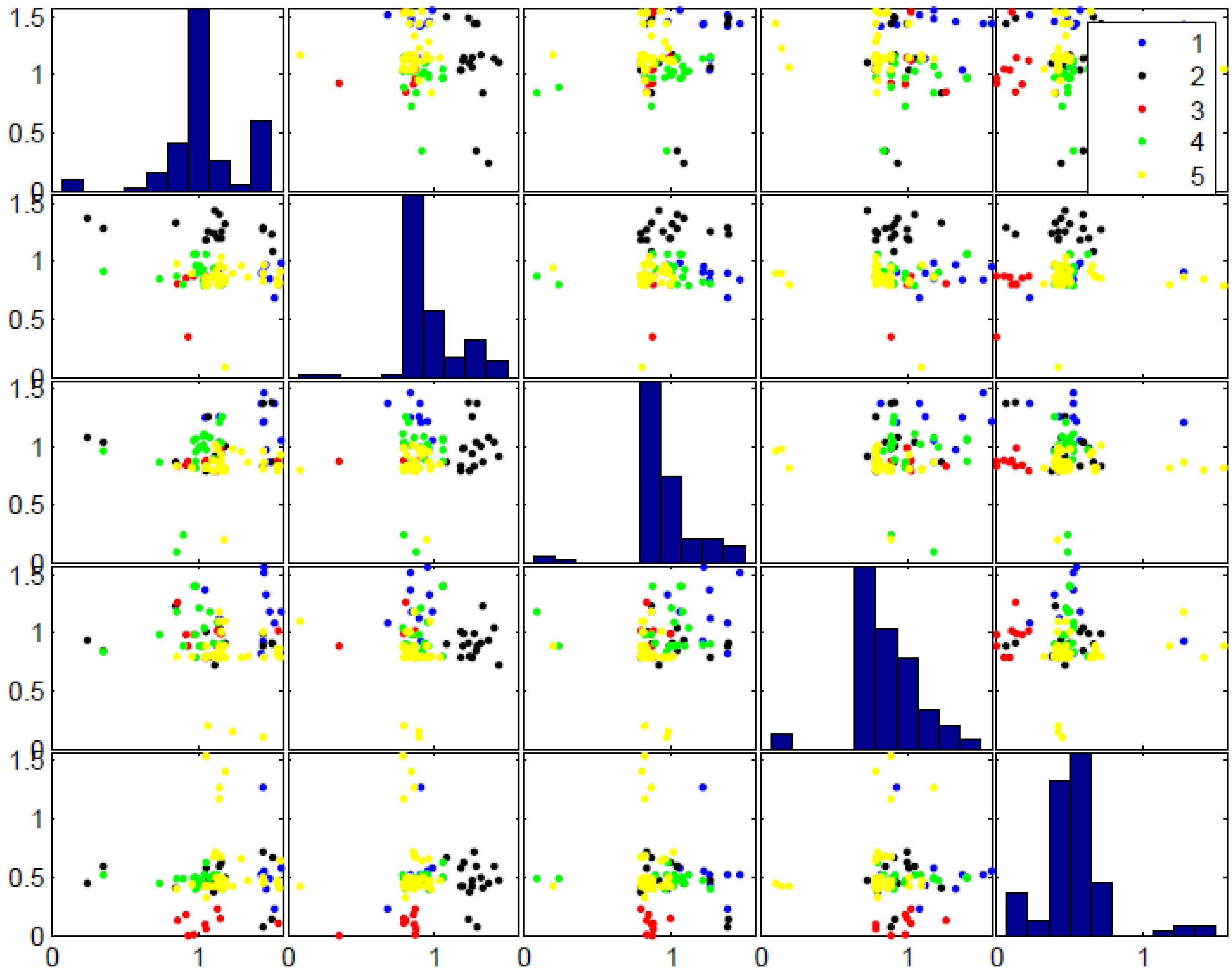
Gen = 4



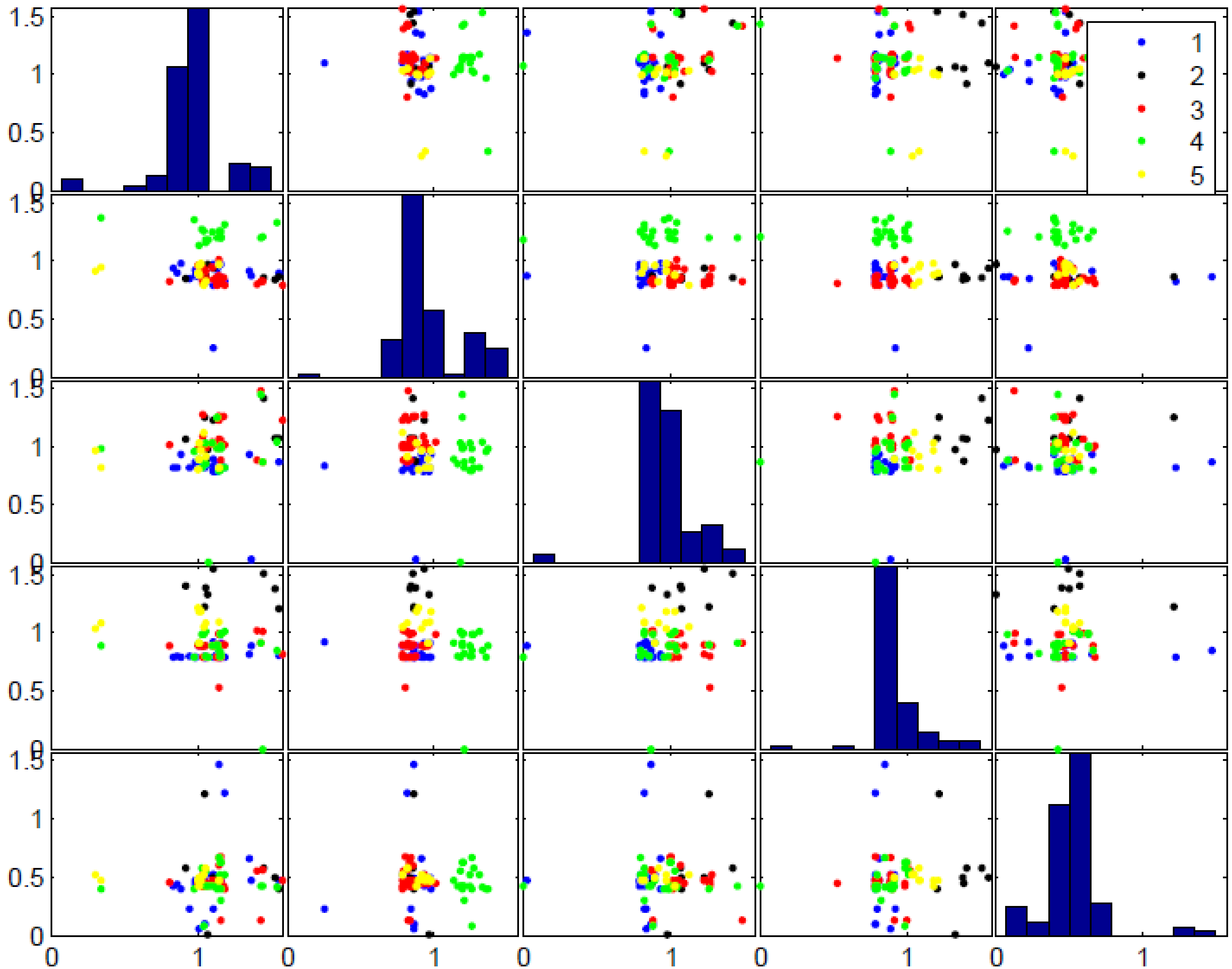
Gen = 5



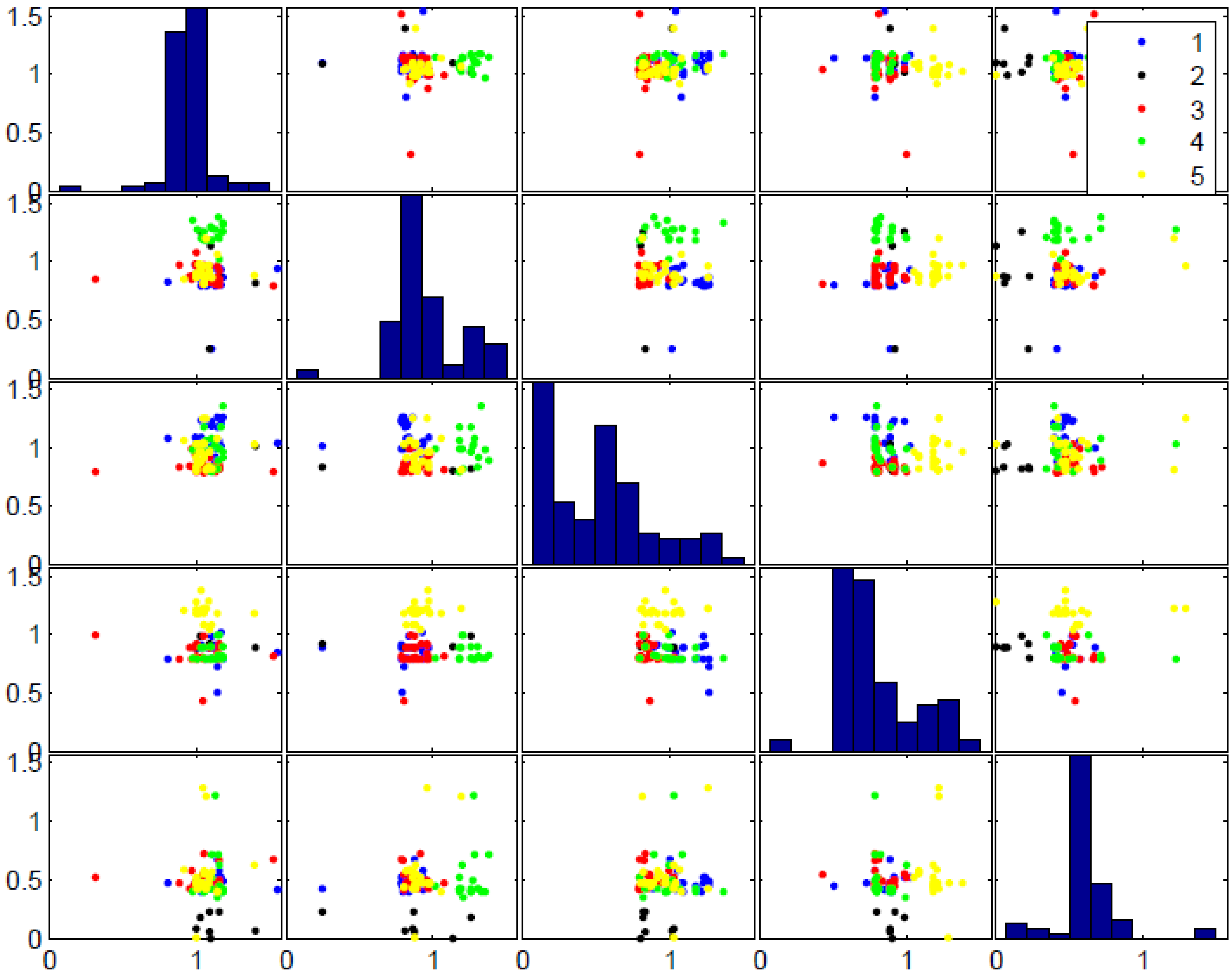
Gen = 6



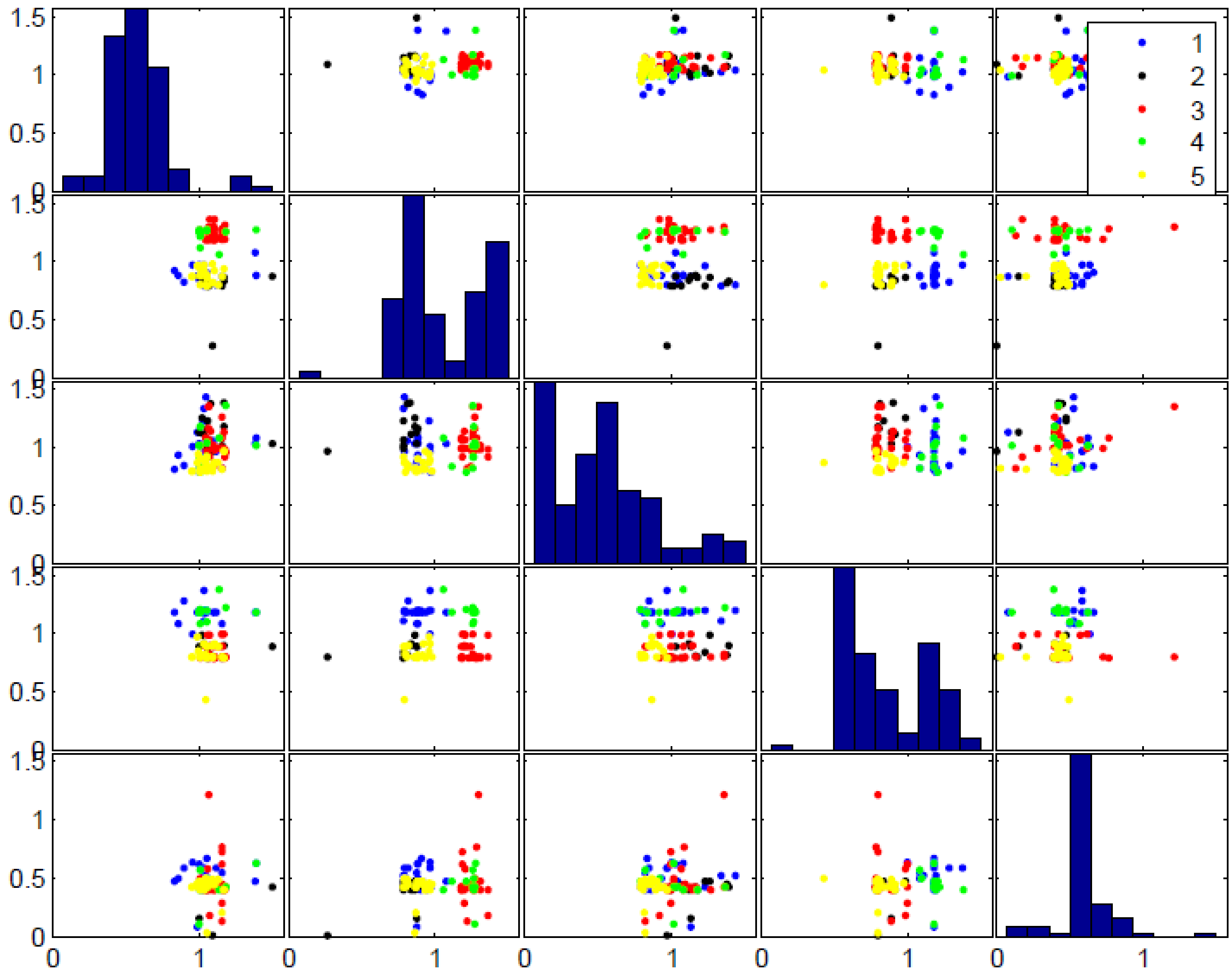
Gen = 7



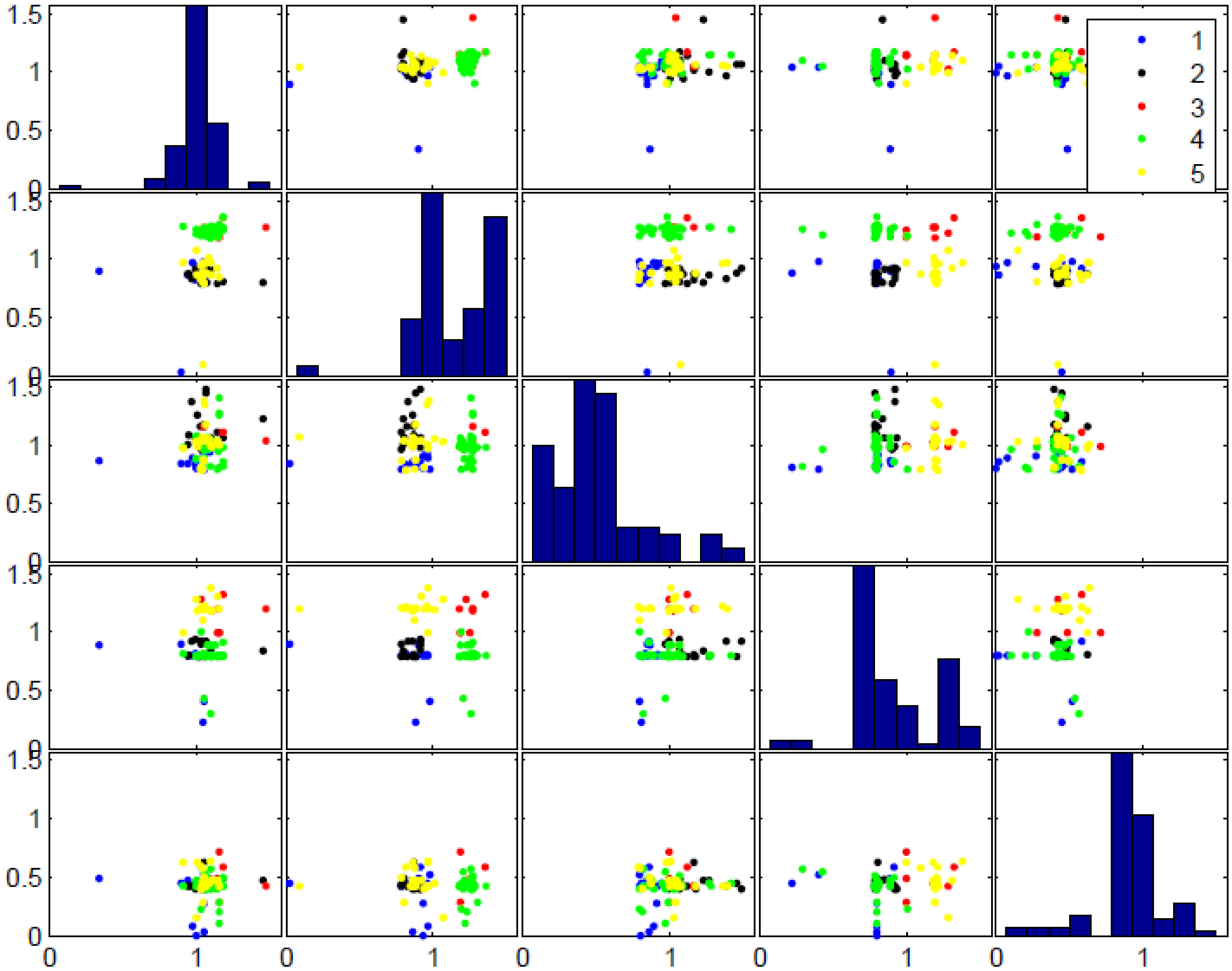
Gen = 8



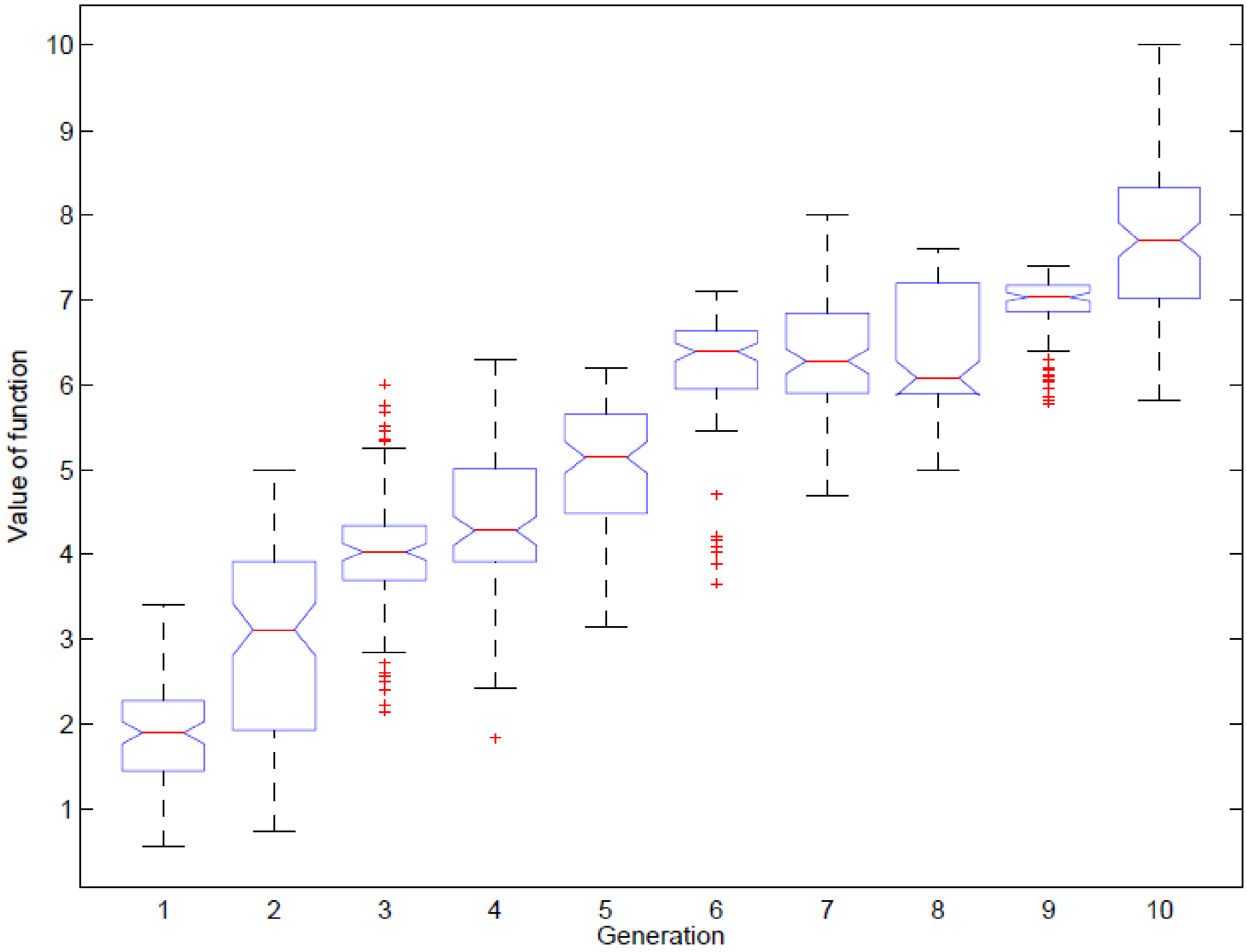
Gen = 9



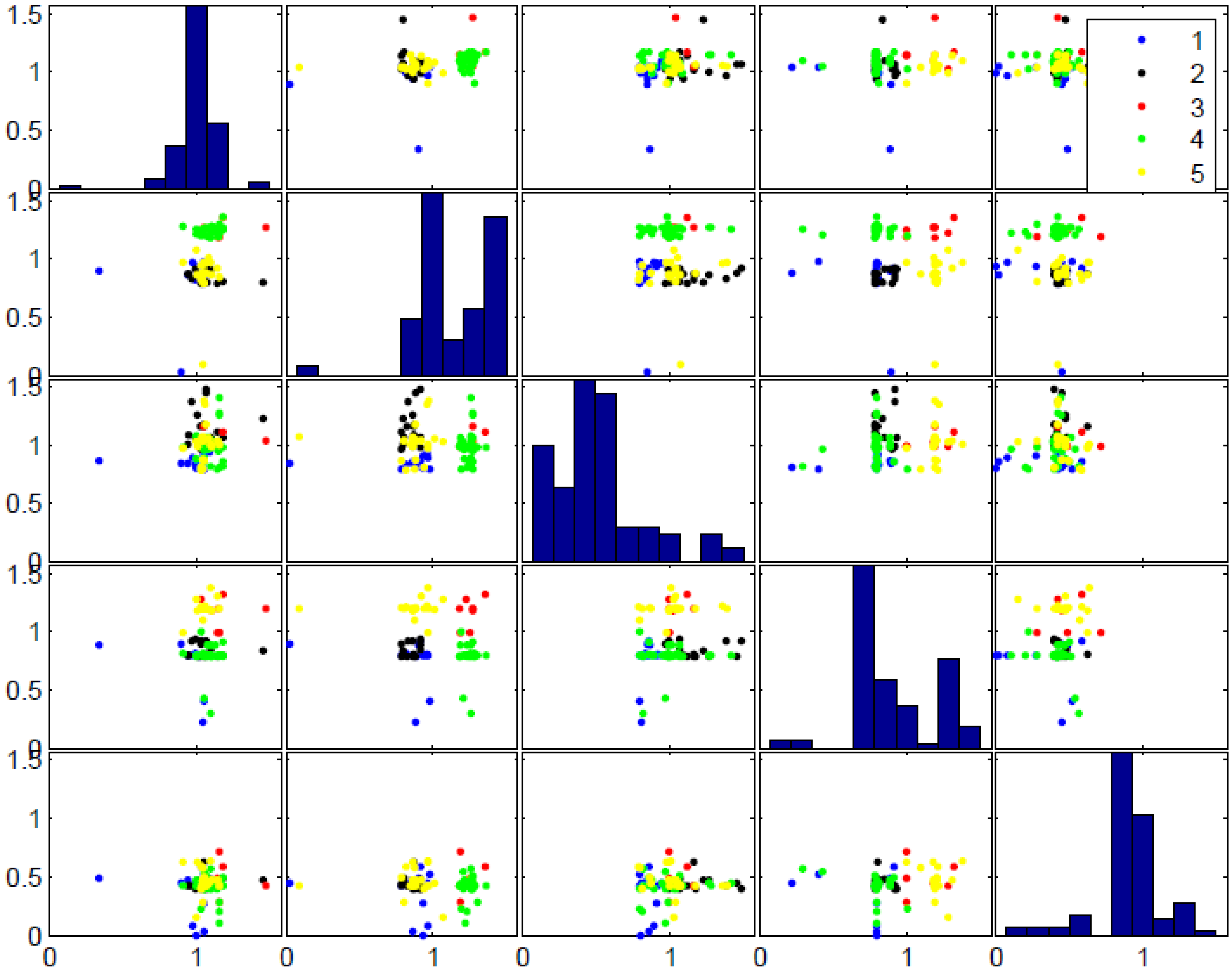
Gen = 10



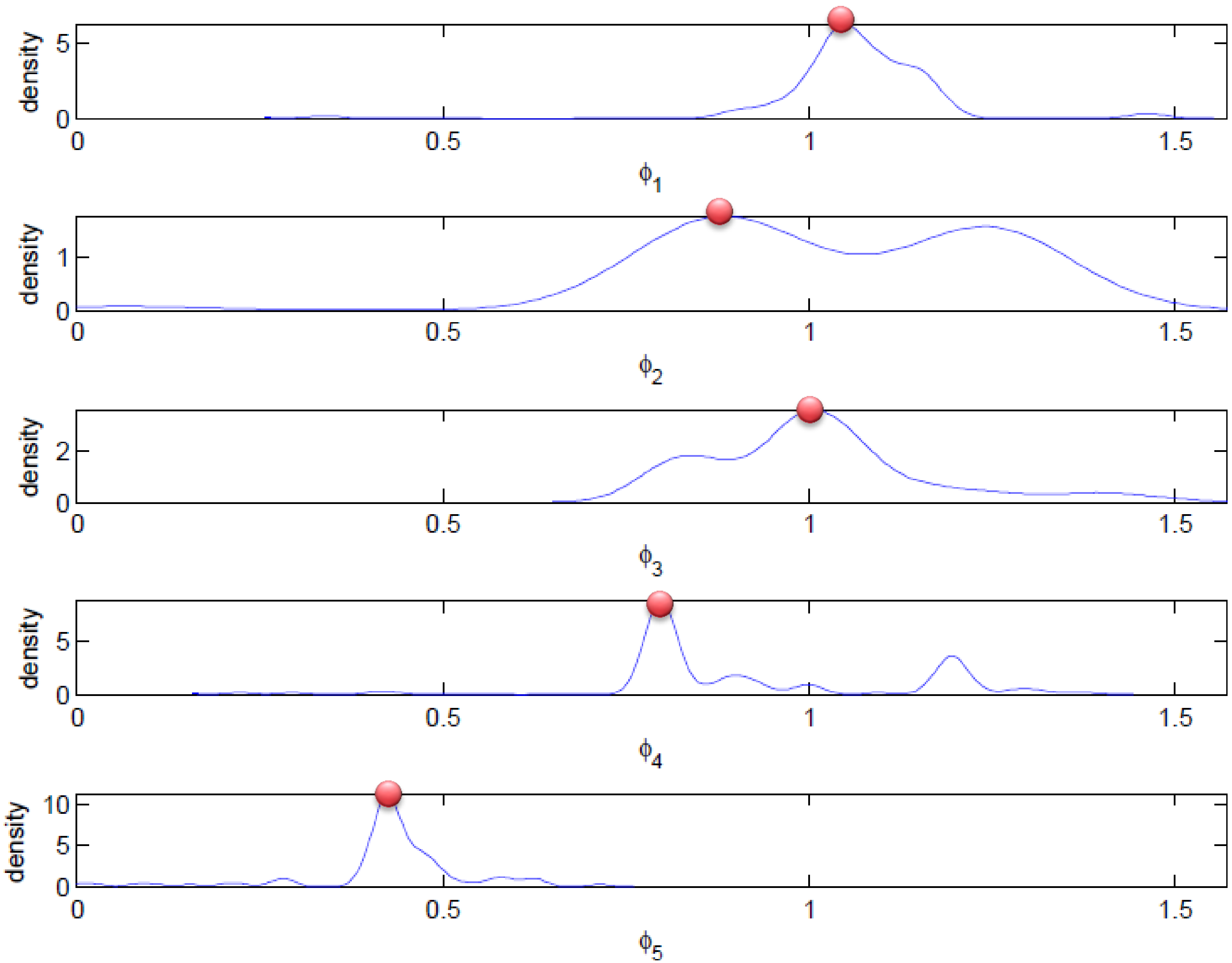
After 10 generations…



Gen = 10

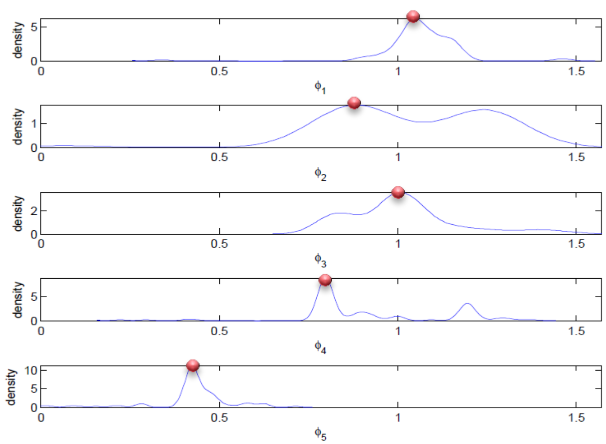


After 10 generations…



After 10 generations…

Convert phi terms to gains as follows



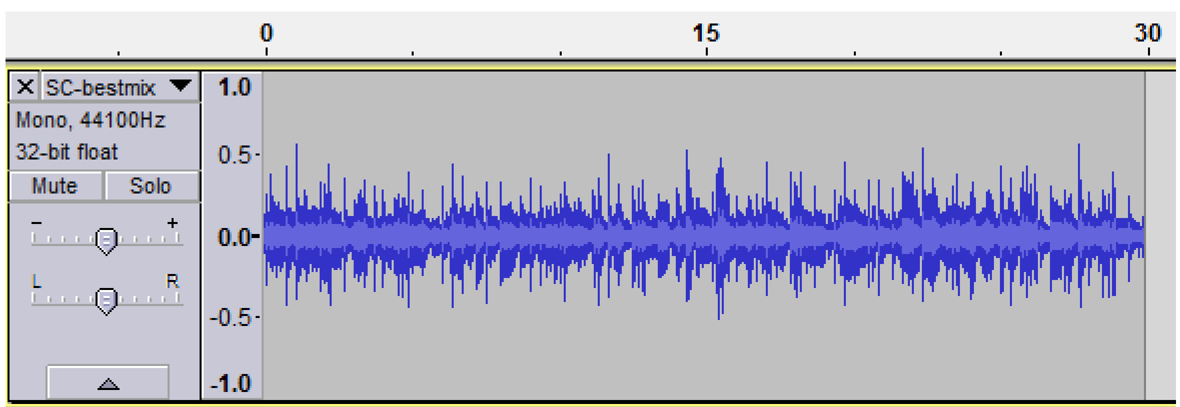
$$g_1 = r\cos\phi_1$$

$$g_j = r\cos\phi_j \prod_{k=1}^{j-1} \sin\phi_k$$

$$g_{n-1} = r\sin\theta \prod_{k=1}^{n-2} \sin\phi_k$$

$$g_n = r\cos\theta \prod_{k=1}^{n-2} \sin\phi_k$$

```
mix = audio*gain';                    %generate mix from audio and gains
```

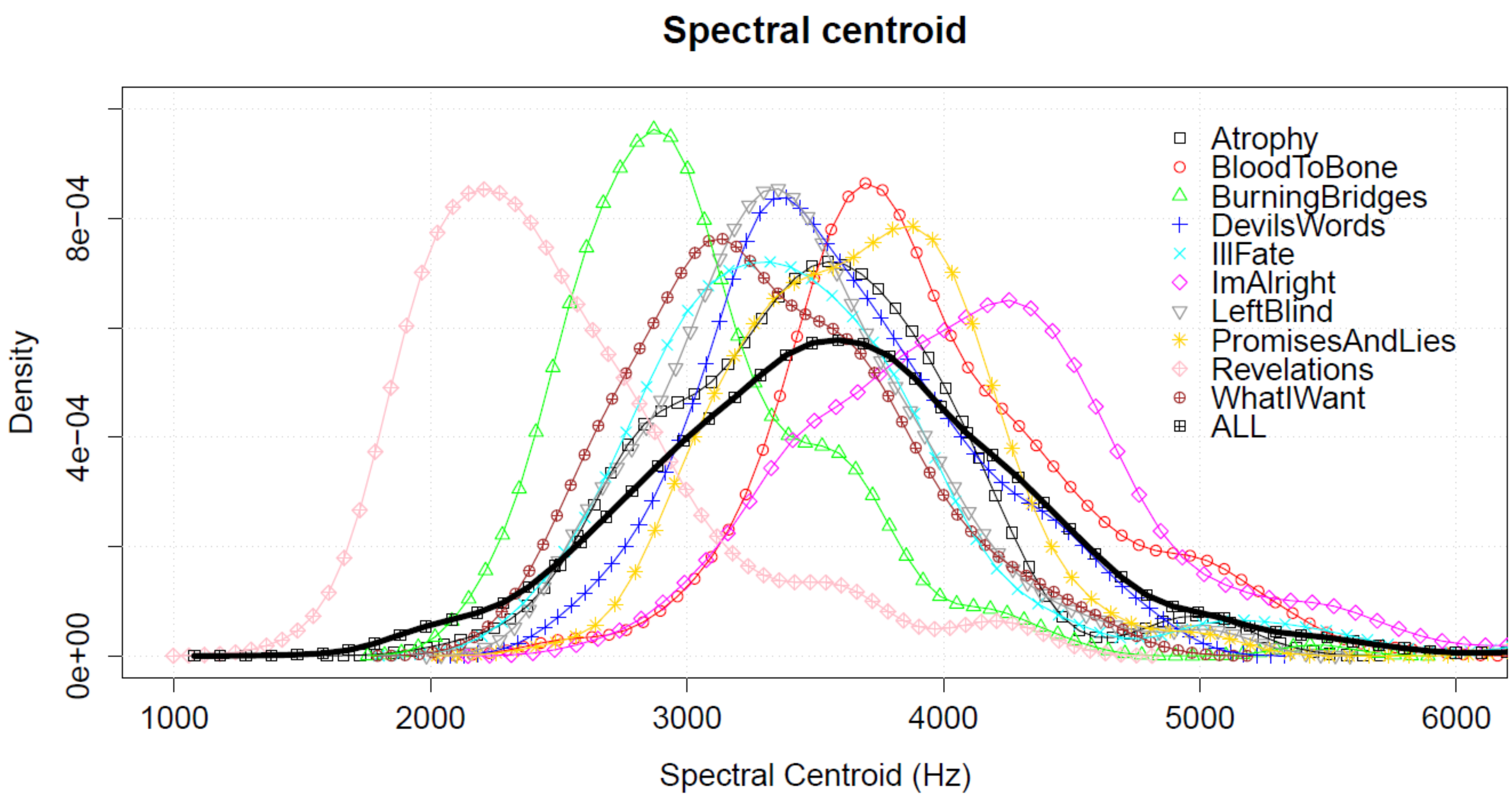


FURTHER WORK

Improvements / optimisation

- Optimise
  - Distribution of initial population
  - Number of clusters
  - Selection of representatives from each cluster
- Test the effects of…
  - GA parameters (Population size, Selection, Crossover, Mutation, …, etc)
- Expand solution space
  - Add equalisation
  - Add stereo panning

Improvements / optimisation



“Variation in multitrack mixes: analysis of low-level audio signal features”,
Wilson/Fazenda, 2016

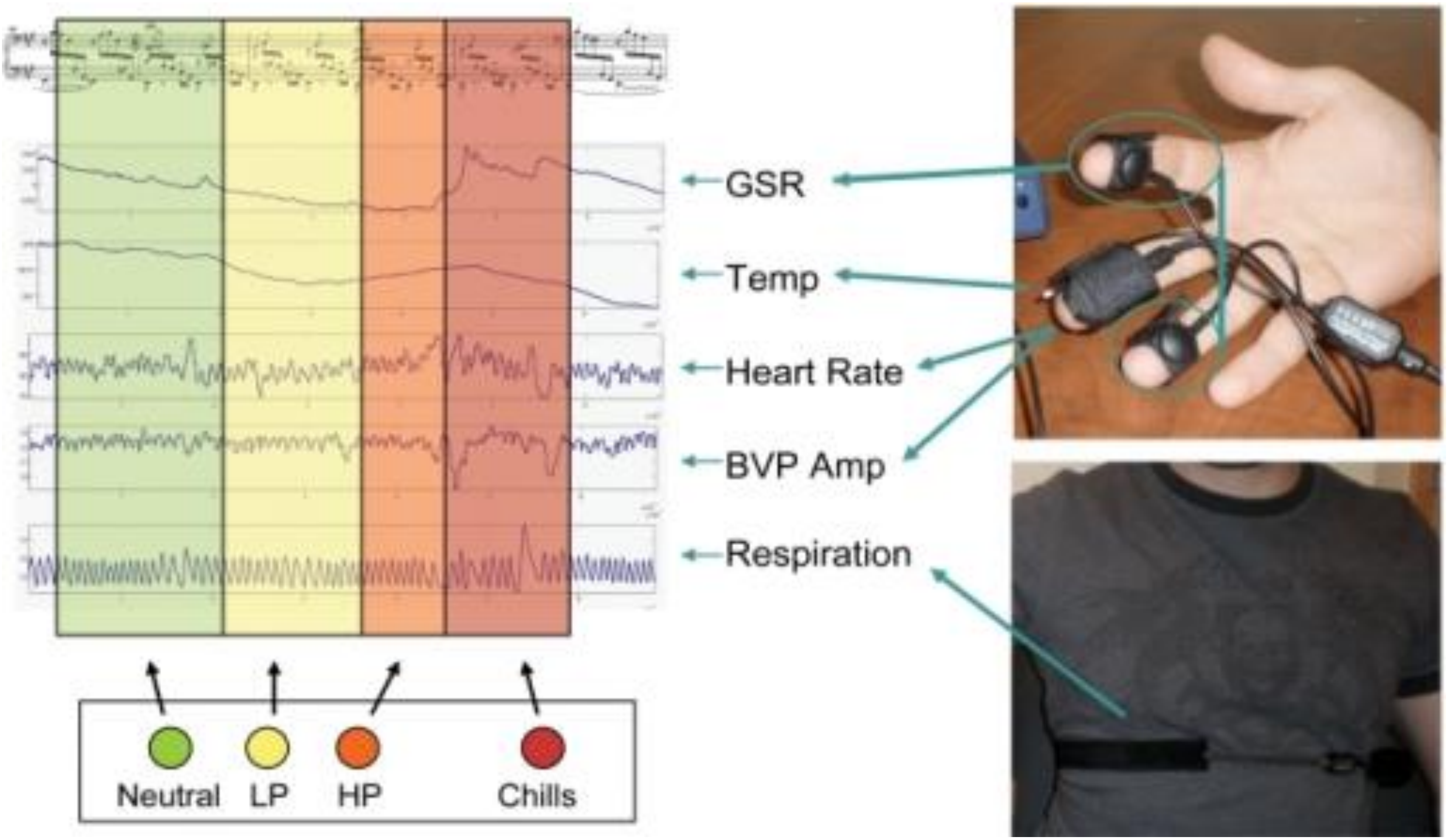
Fitness function

Subjective (ratings):

- “How much do you like this mix?”
  - Creates mixes you would like
- “How clear is this mix?”
  - Creates mixes that are clear
- “How exciting is this mix?”
  - Creates mixes that are exciting

Fitness function

Subjective (physiological):



Conclusions

- Development of system which learns how to present object-based audio according to qualities desired by the user.
- Can be used to learn how certain qualities of audio/music are perceived
- Can be expanded to included further processing

Thanks for listening

Alex Wilson
Postgraduate Researcher | Acoustics Research Centre
Newton Building, University of Salford, Salford M5 4WT
a.wilson1@edu.salford.ac.uk http://www.salford.ac.uk
@AlexMacLiam