Subspace clustering on static datasets and dynamic data streams using bio-inspired algorithms

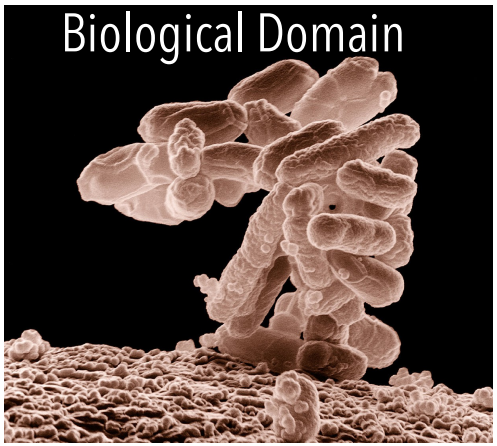
Sergio Peignier
Supervisors: Christophe Rigotti and Guillaume Beslon

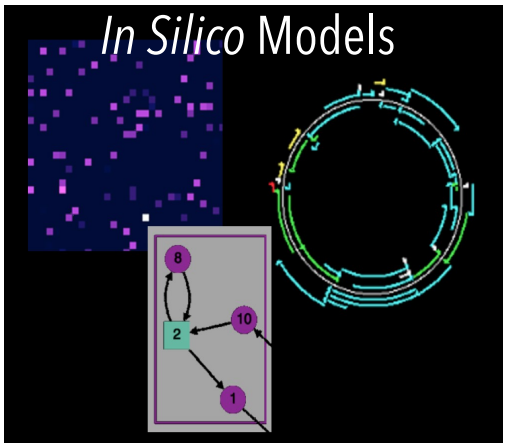
Université de Lyon
INSA-Lyon, CNRS, INRIA
LIRIS, UMR5205
F-69621, France

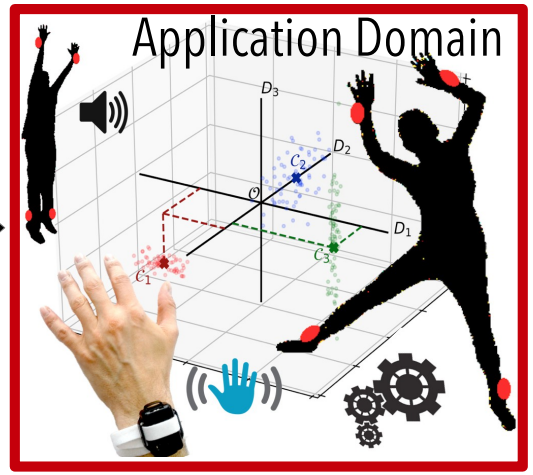
Context

EvoEvo project

- Living organisms have **evolved** mechanisms to cope with **complex** and **changing environments**.
- Micro-organisms mainly rely on evolution.
- Mechanism: **Evolvable genome structure**.
- Application: **Subspace clustering** (data mining).







Hypothesis

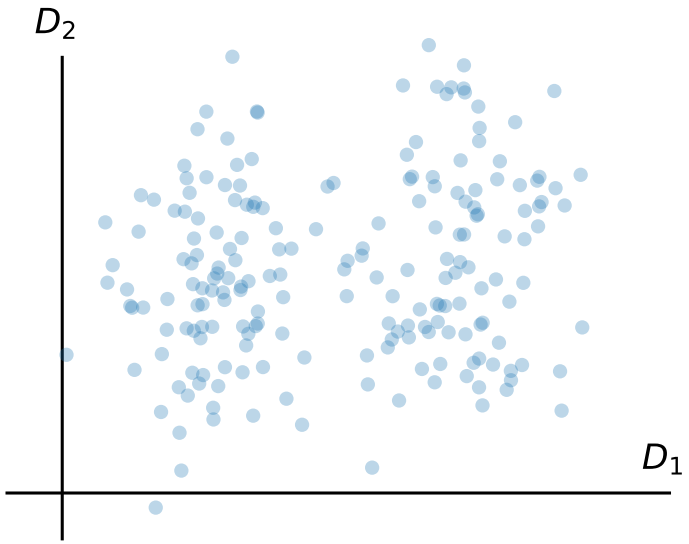
It is possible to **take advantage** of an **evolvable genome structure** to tackle the **subspace clustering** task.

Outline

Outline

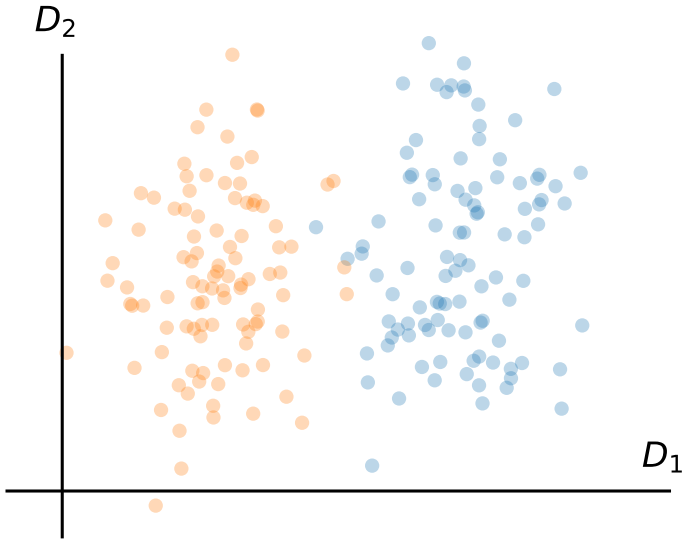
Clustering

- **Data objects** exist in a **space** $\mathcal{D} = \{D_1, D_2 \dots\}$

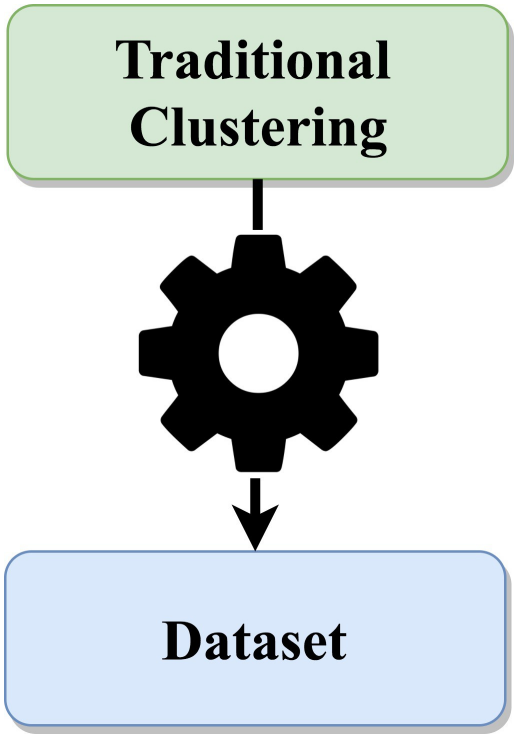


Clustering

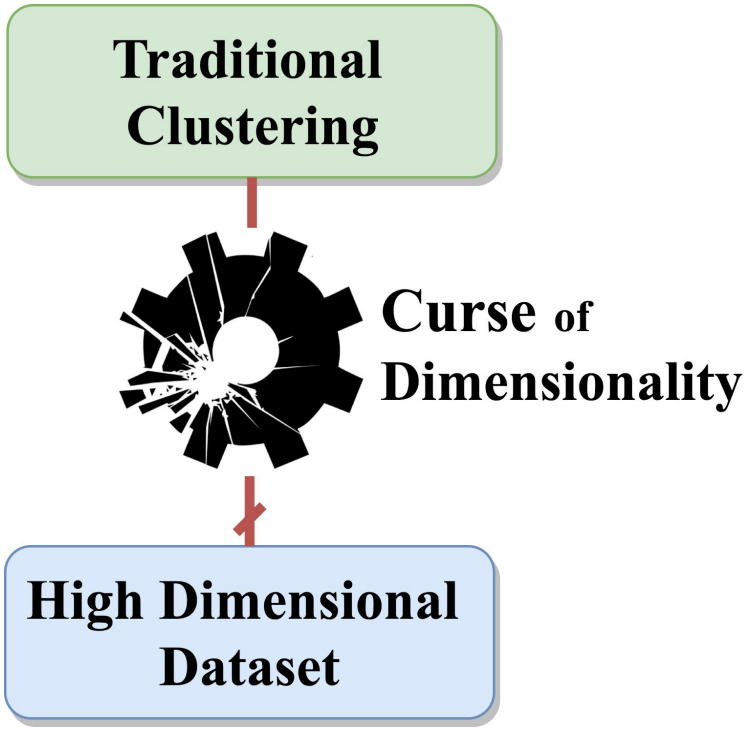
- **Data objects** exist in a **space** $\mathcal{D} = \{D_1, D_2 \dots\}$
- **Dataset** $\mapsto \{$**Cluster**$\}$: **Data objects** in the **same** cluster are more **similar** than objects from **different** ones.



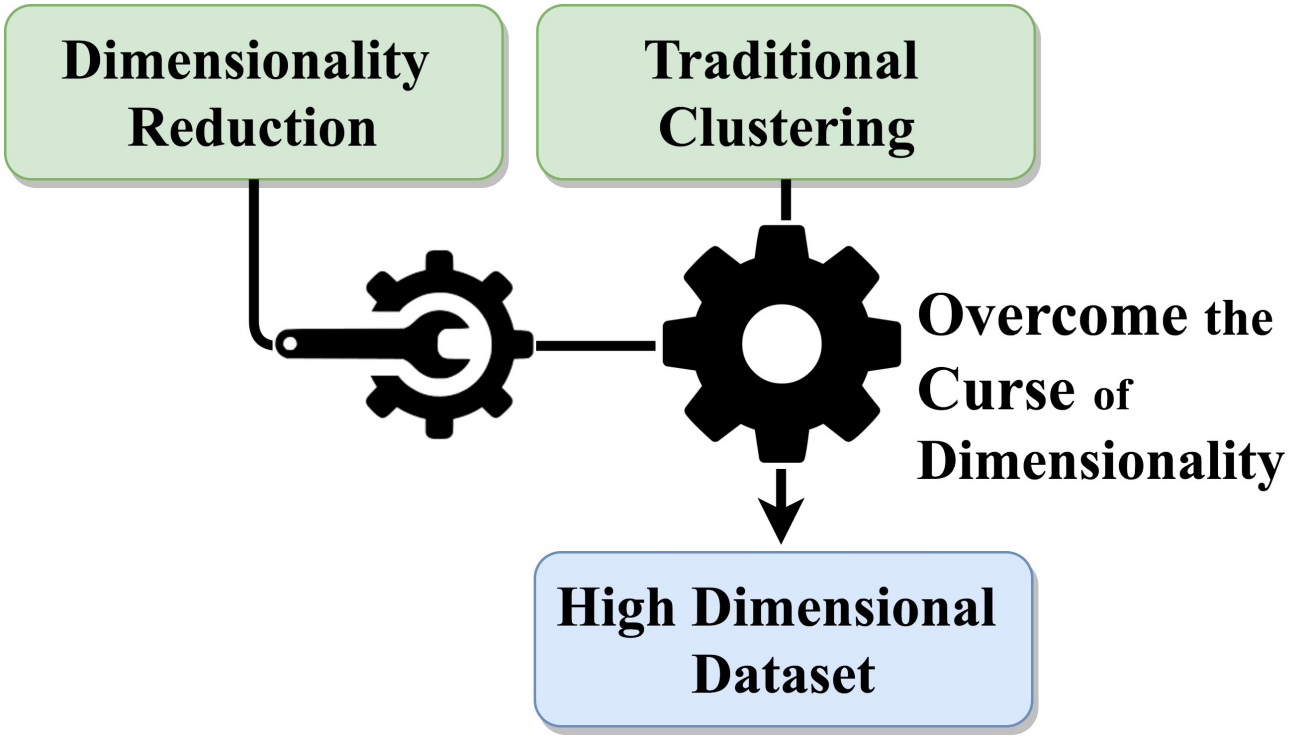
Overcoming the curse of dimensionality



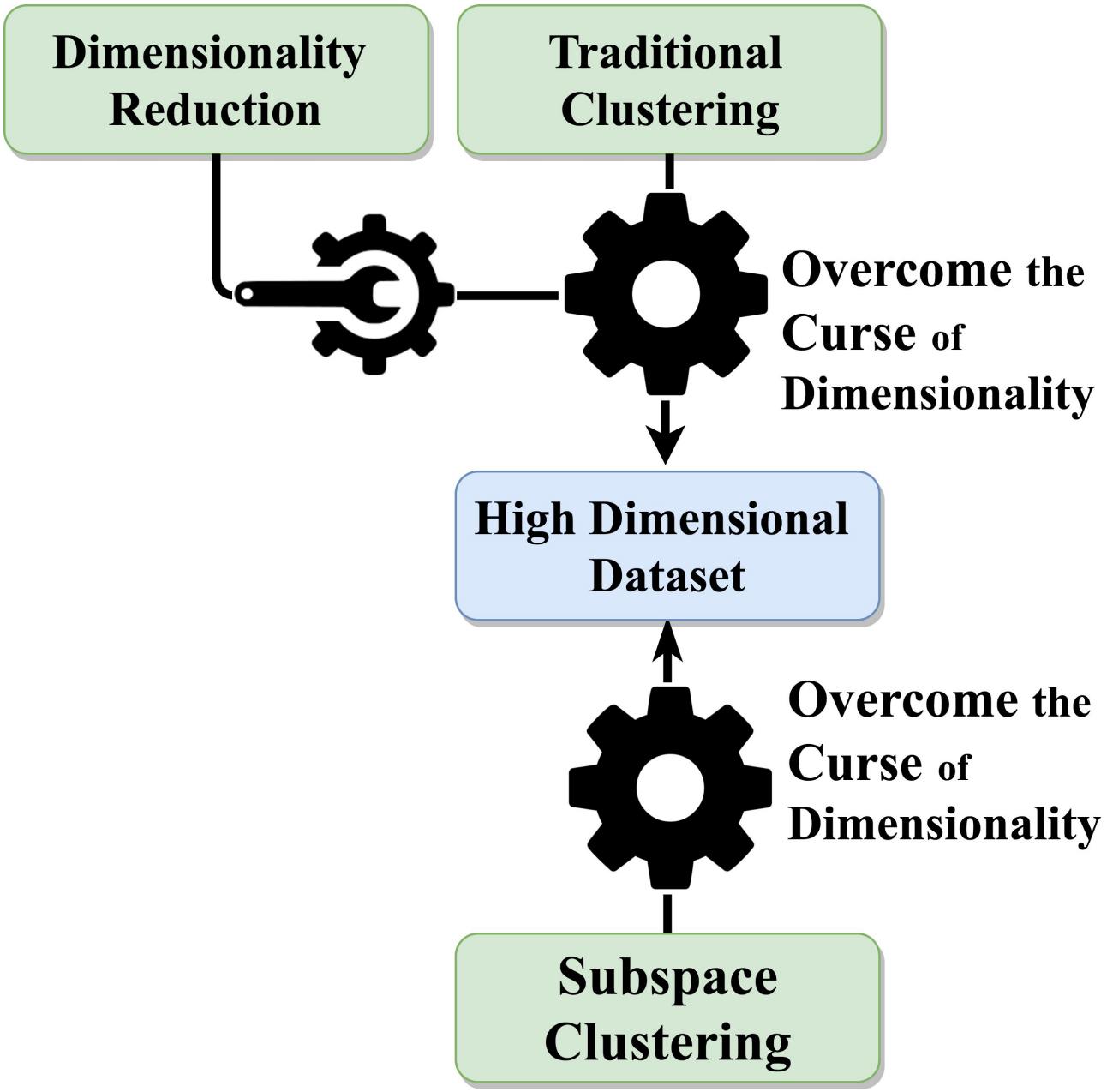
Overcoming the curse of dimensionality



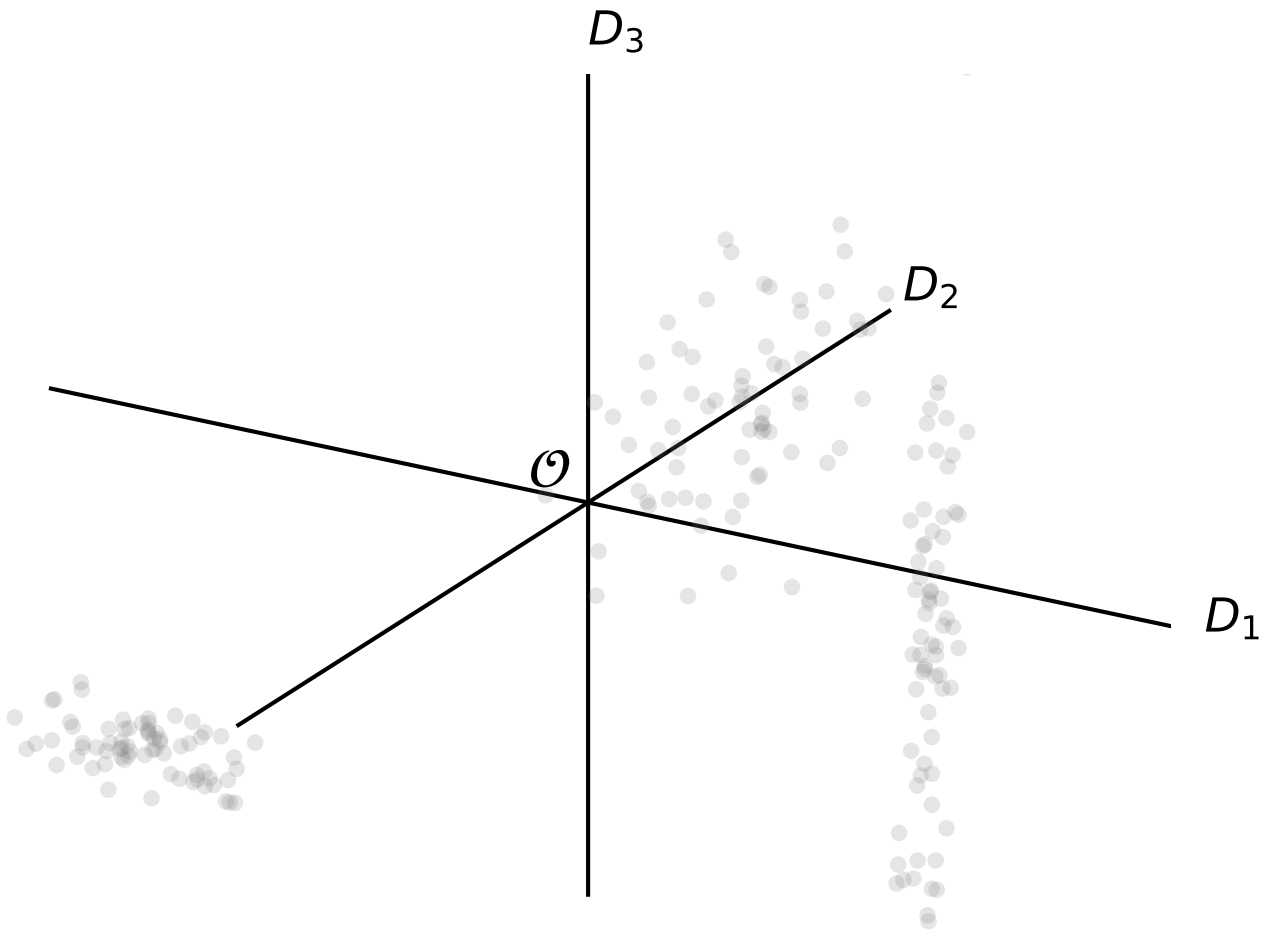
Overcoming the curse of dimensionality



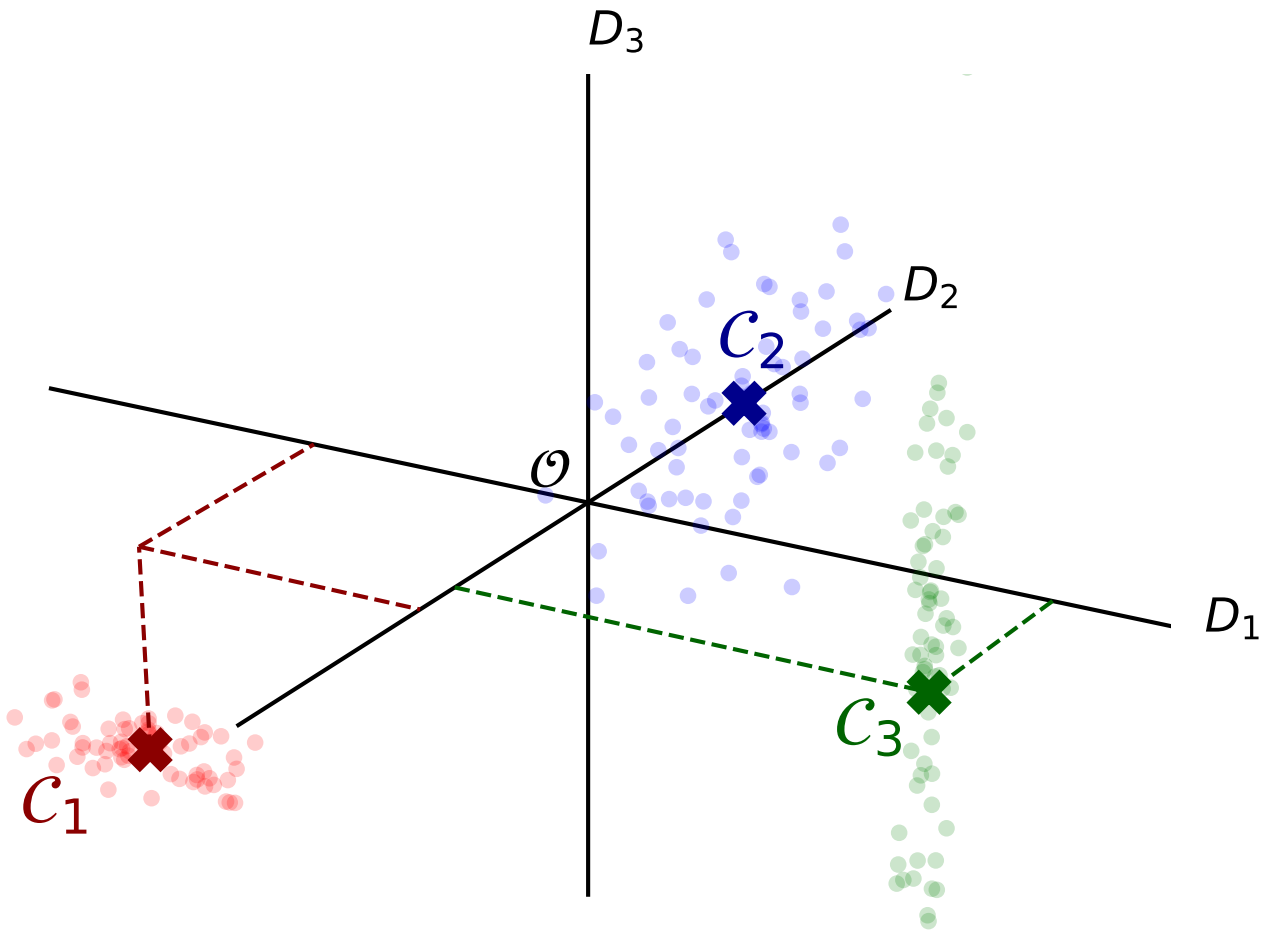
Overcoming the curse of dimensionality



Subspace clustering
[Kriegel et al. ACM TKDD 2009]

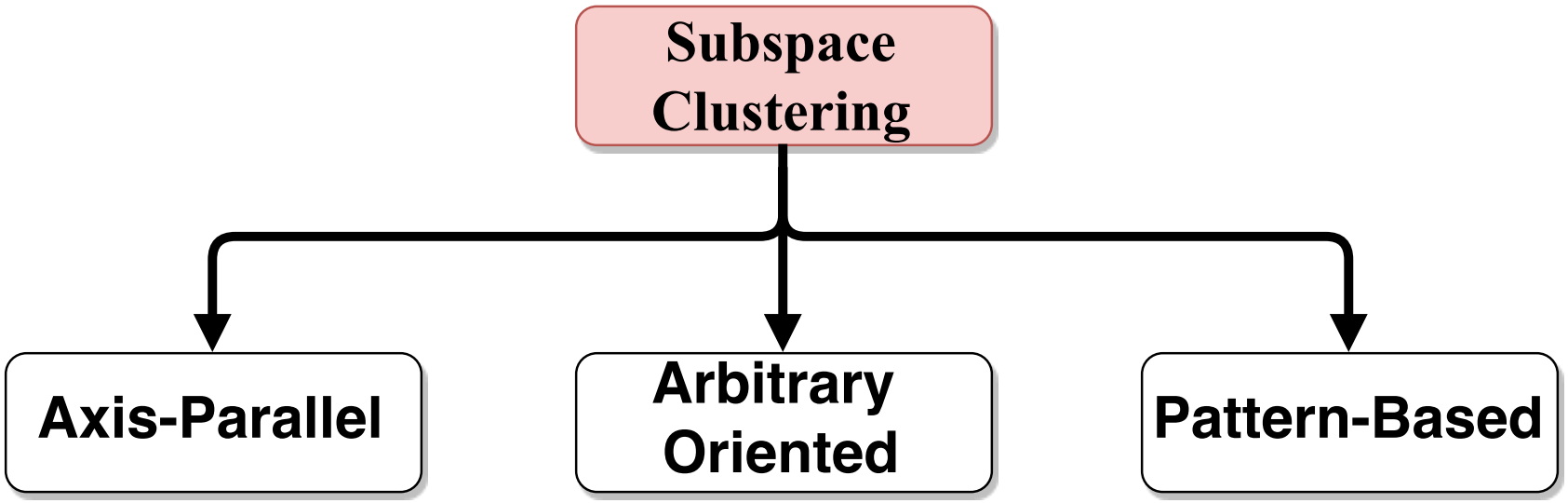


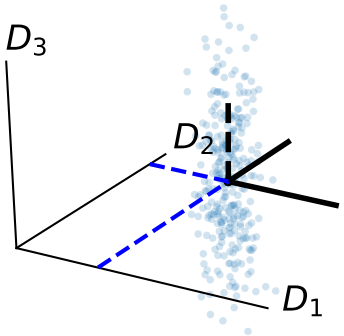
Subspace clustering
[Kriegel et al. ACM TKDD 2009]

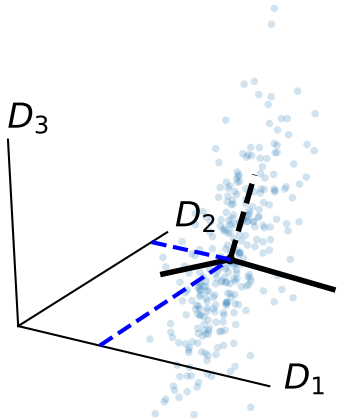


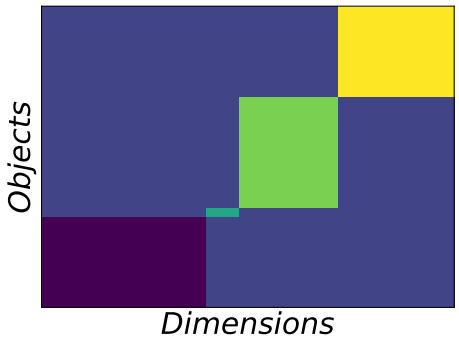
- Dataset $\mapsto$ { ($Clust_1$, $Subspace_1$), ($Clust_2$, $Subspace_2$), ... }.
- Different clusters may be defined in different subspaces.

Subspace clustering families
[Kriegel et al. ACM TKDD 2009]

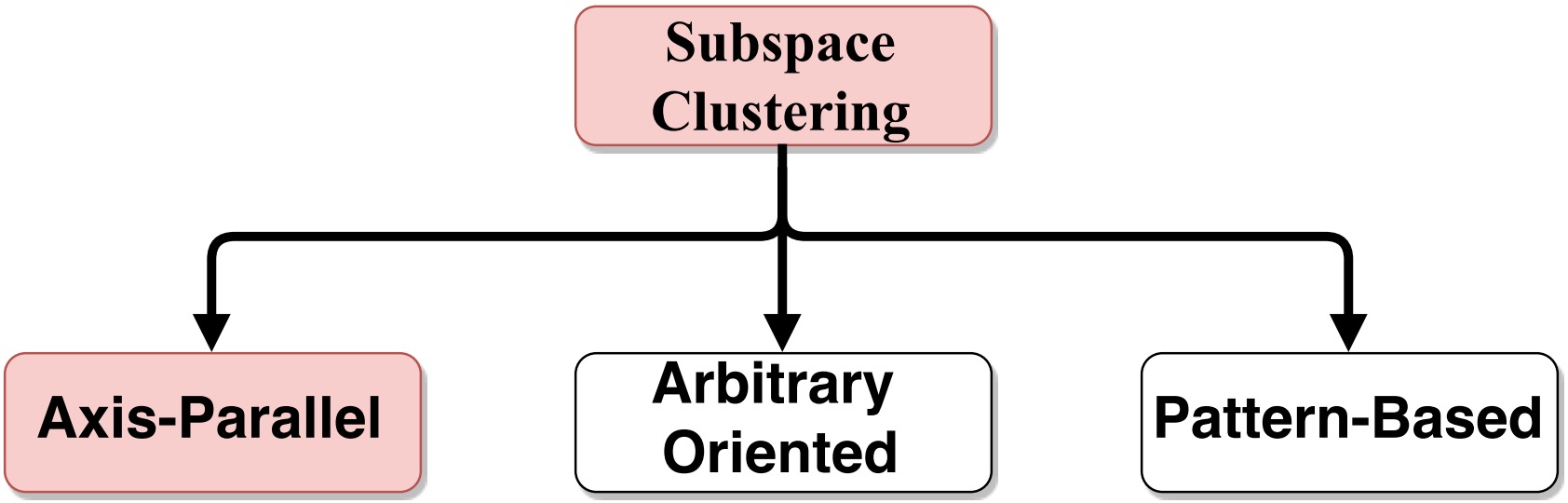


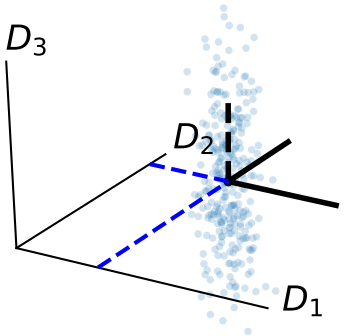


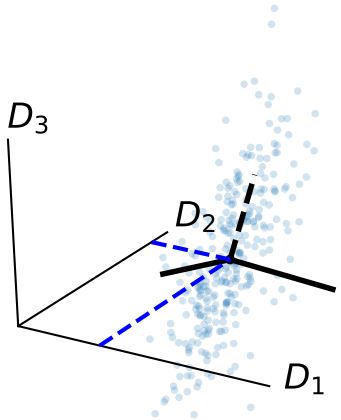


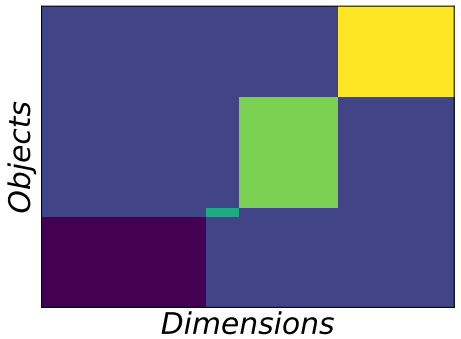


Subspace clustering families
[Kriegel et al. ACM TKDD 2009]

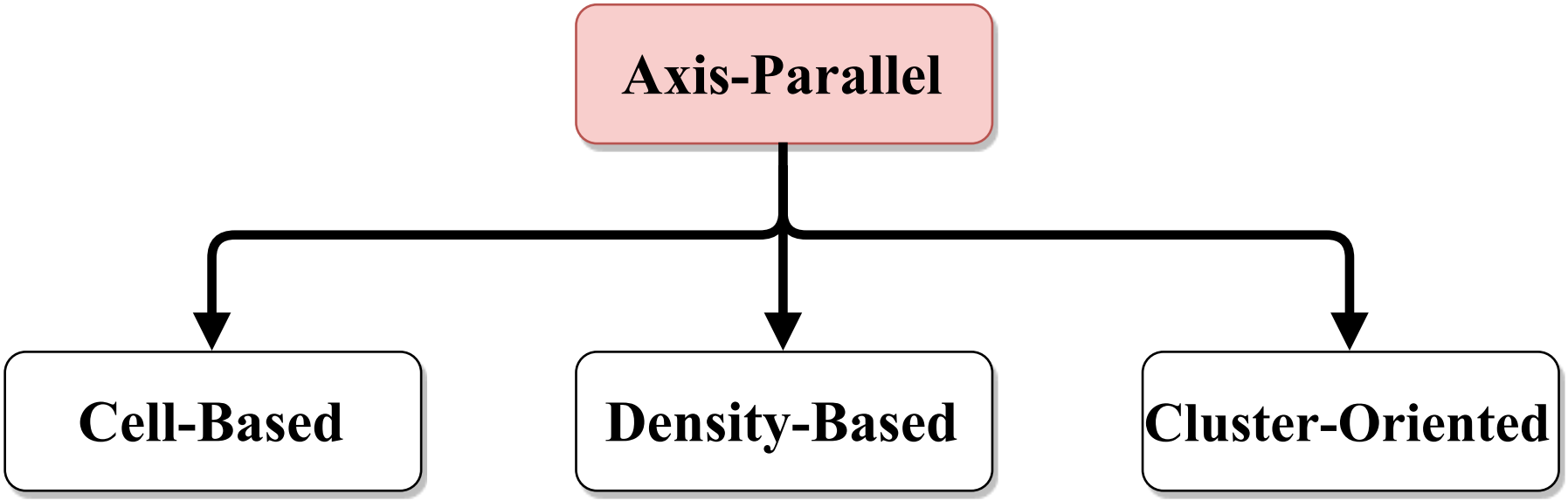


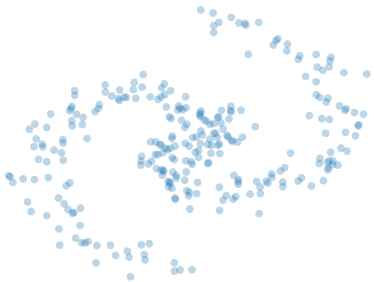


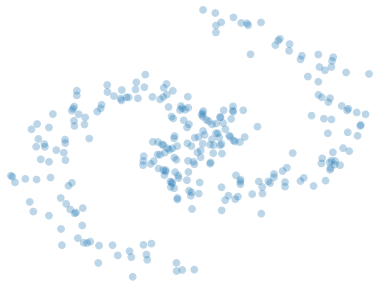


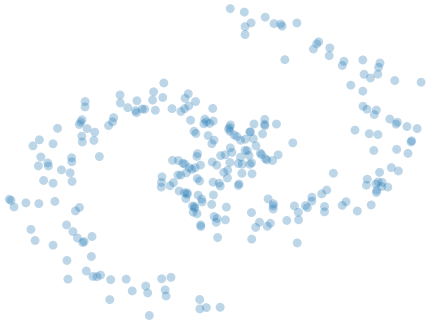


Axis-parallel subspace clustering
[Müller et al. VLDB 2009]

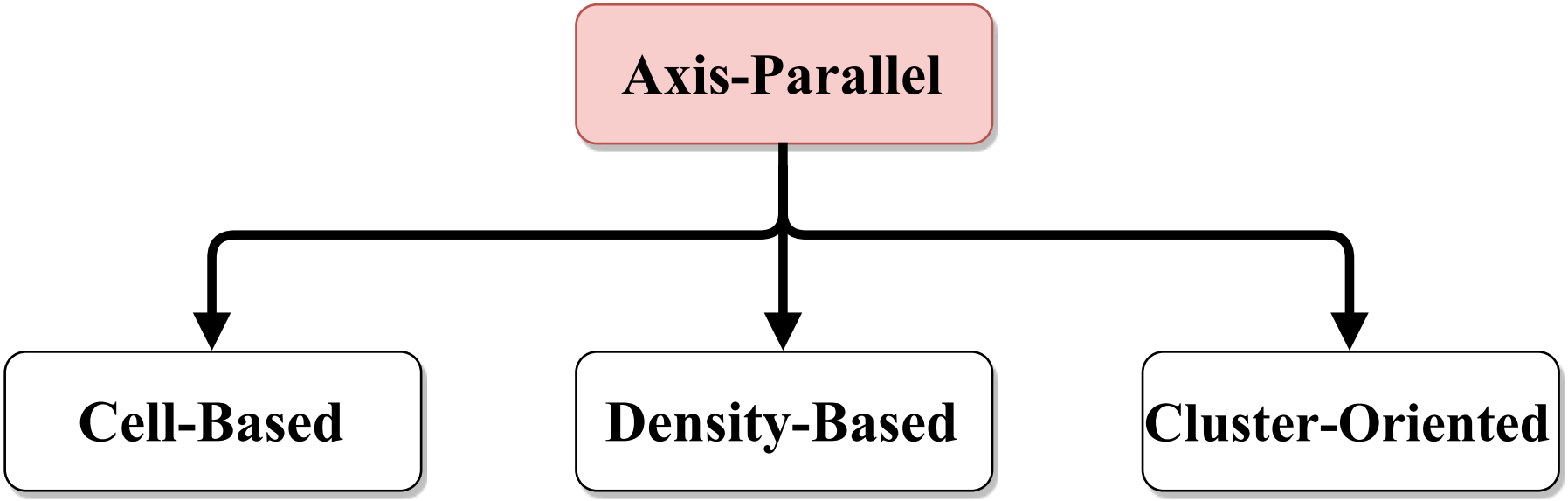




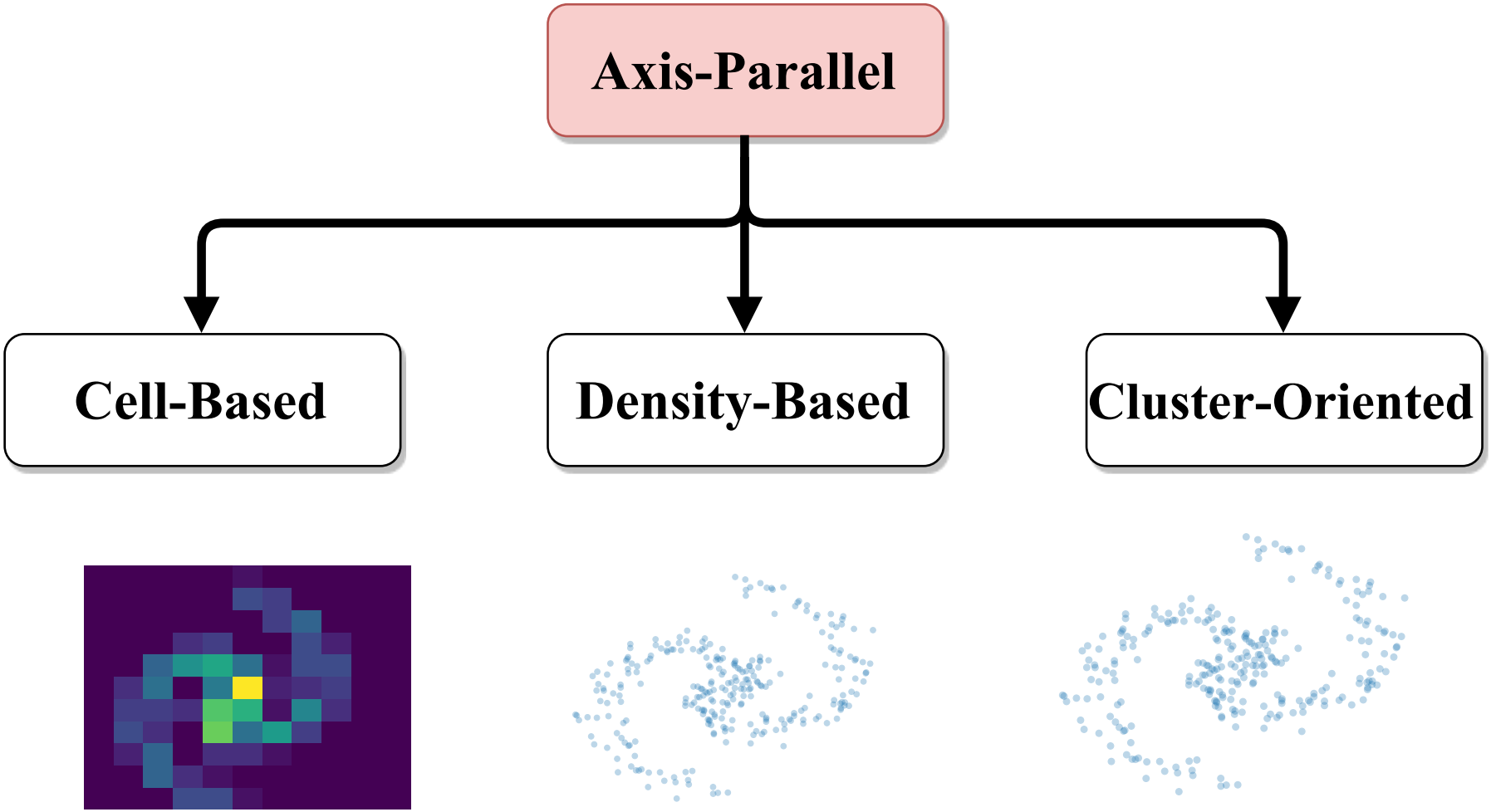




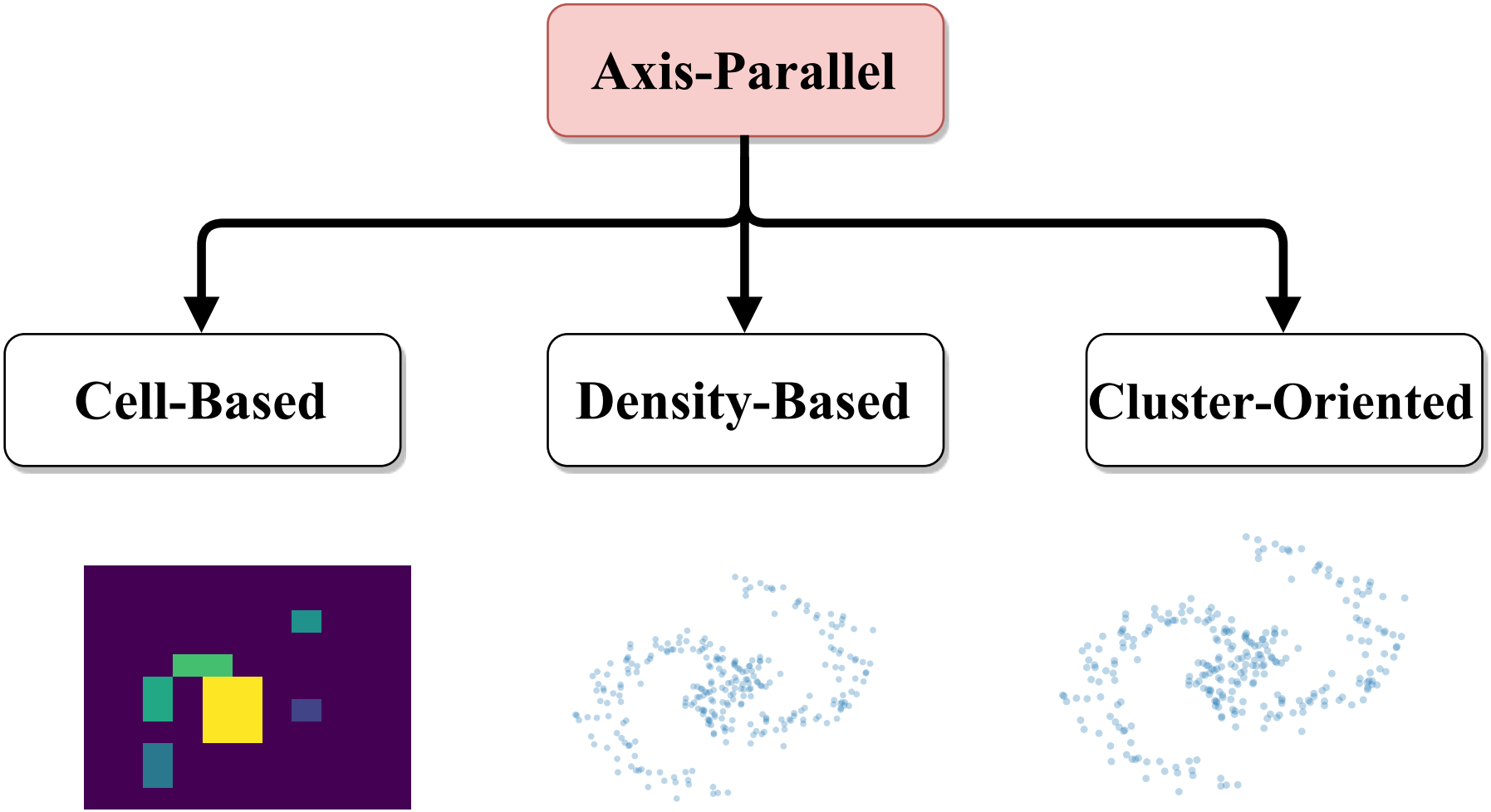
Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



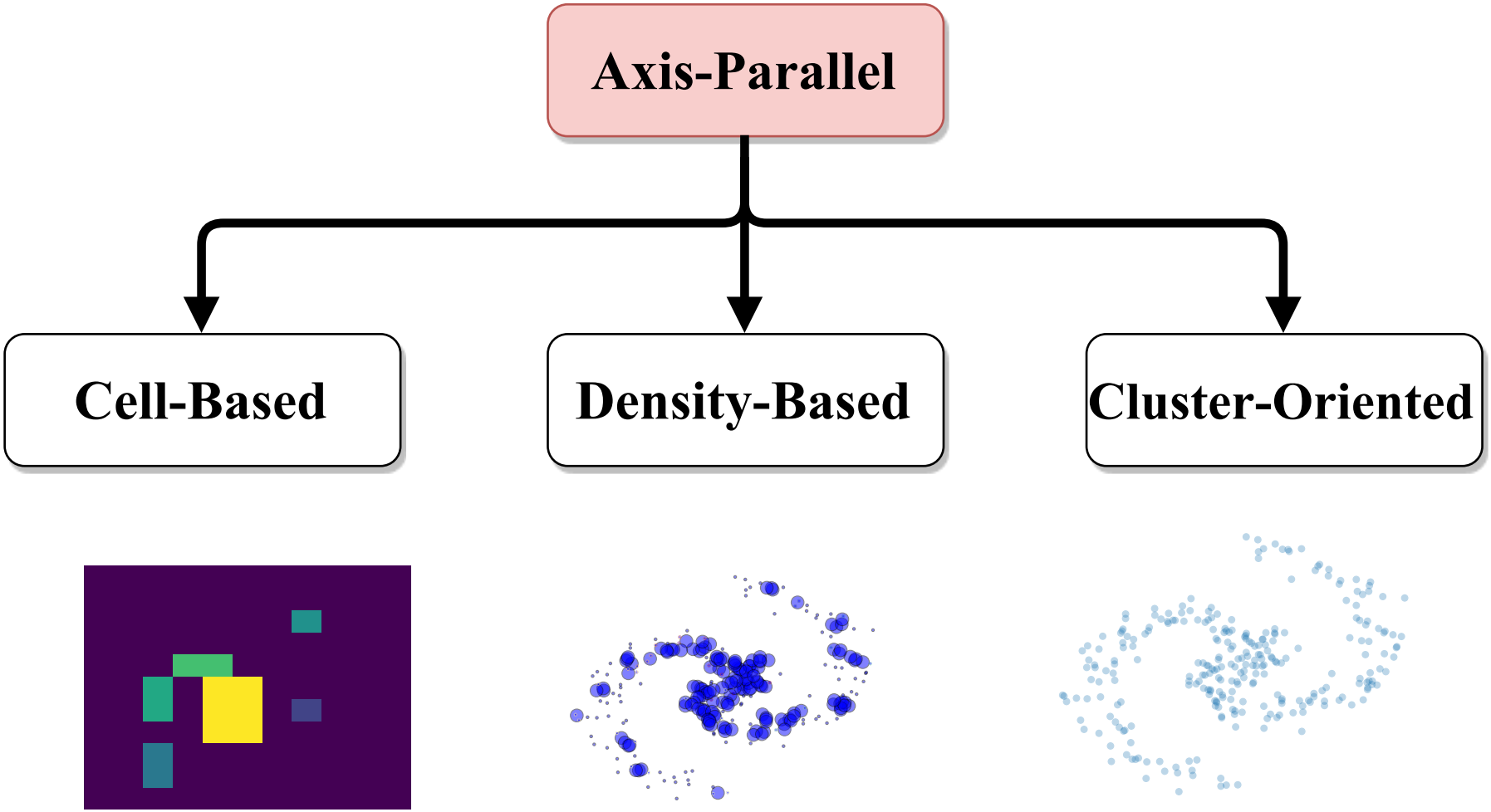
Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



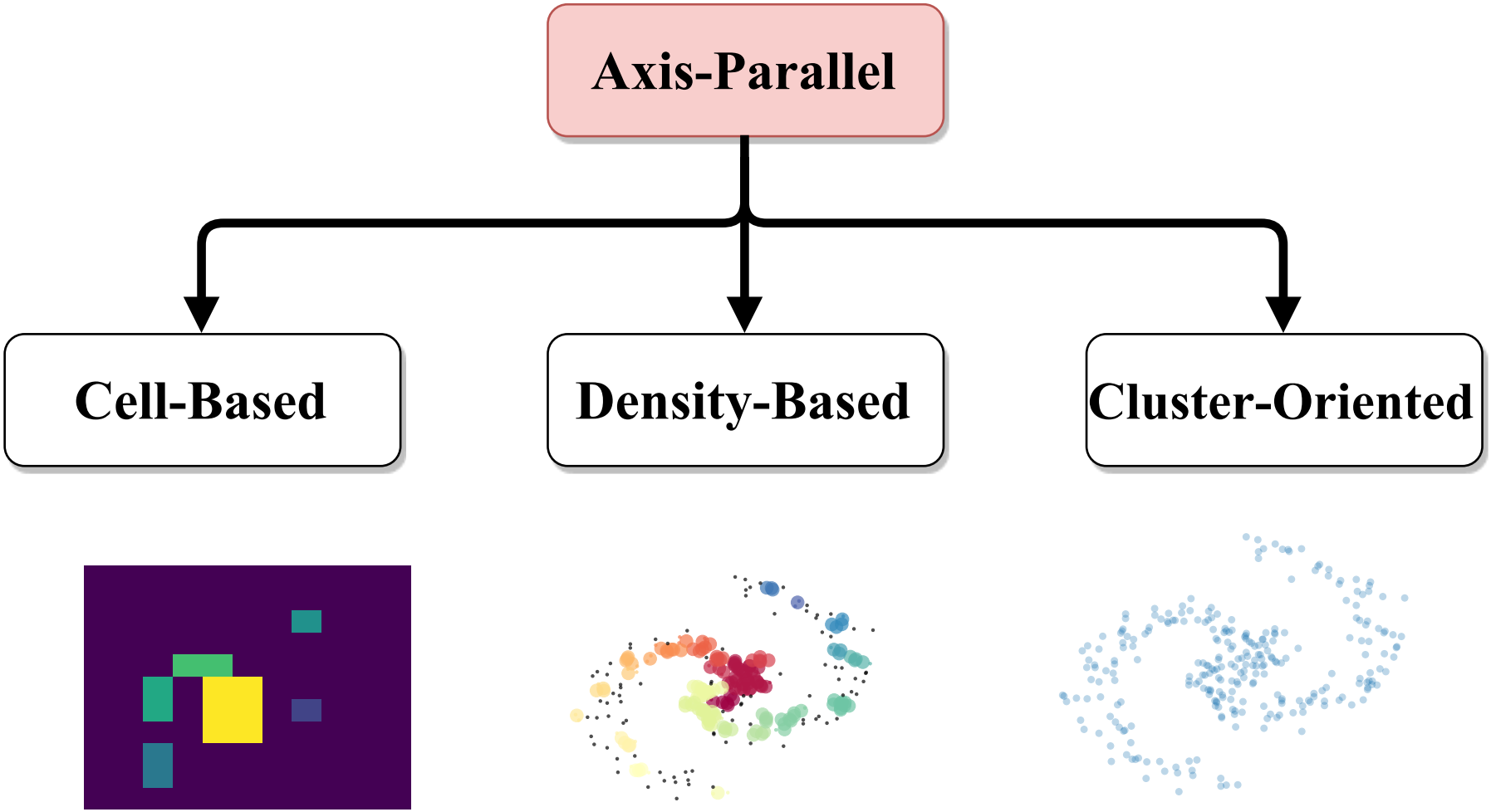
Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



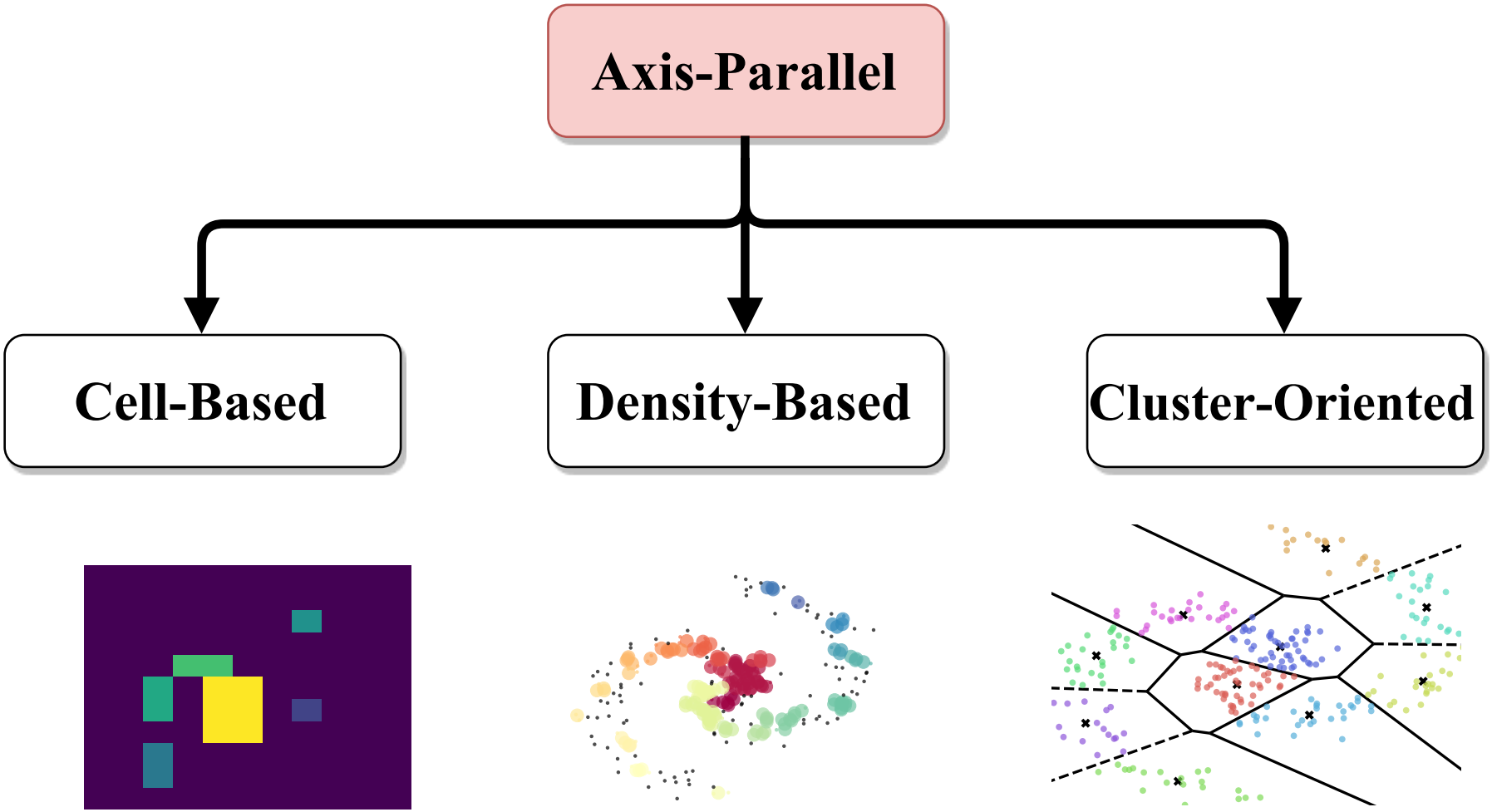
Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



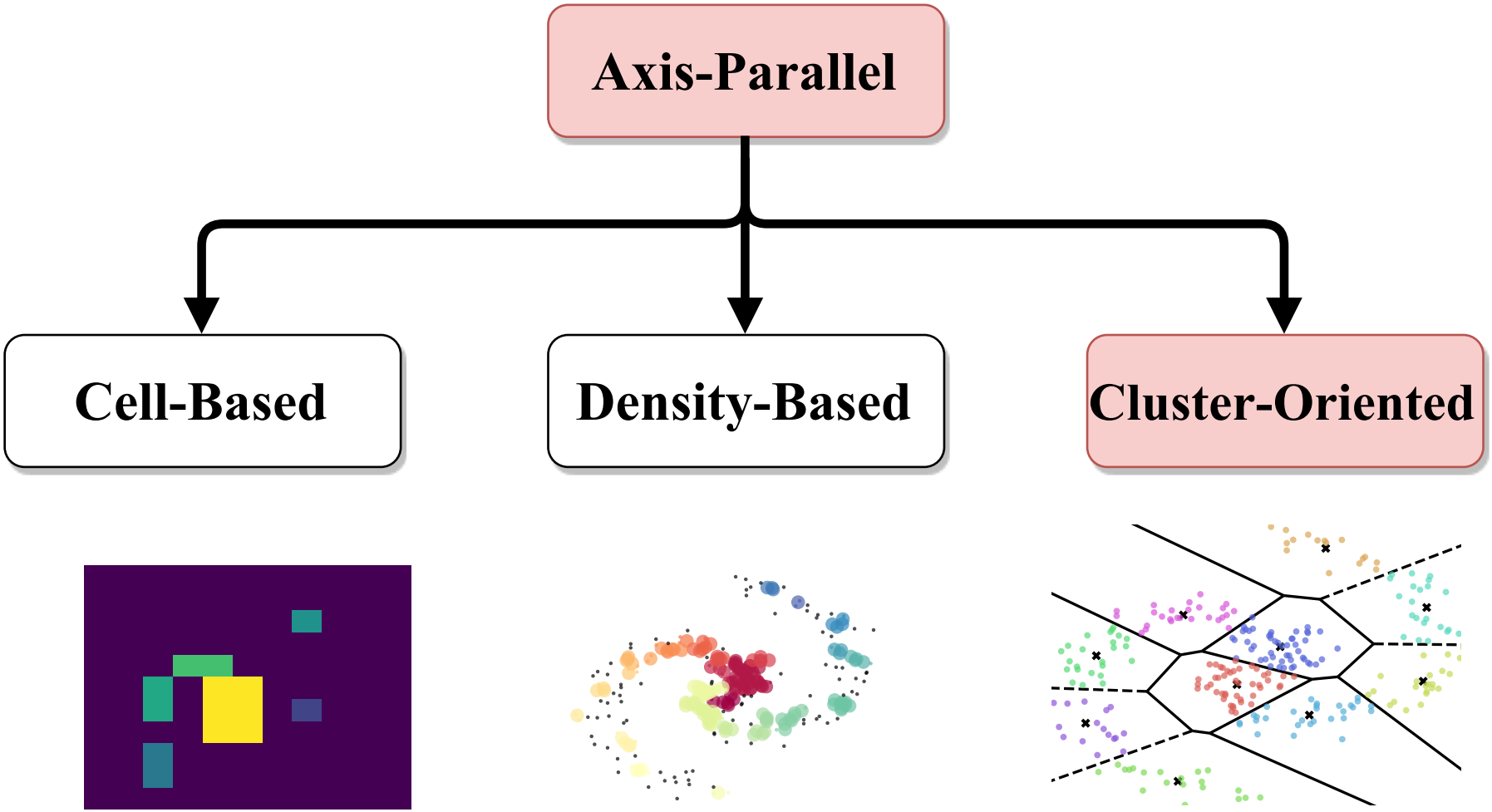
Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



Axis-parallel subspace clustering
[Müller et al. VLDB 2009]



Outline

Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

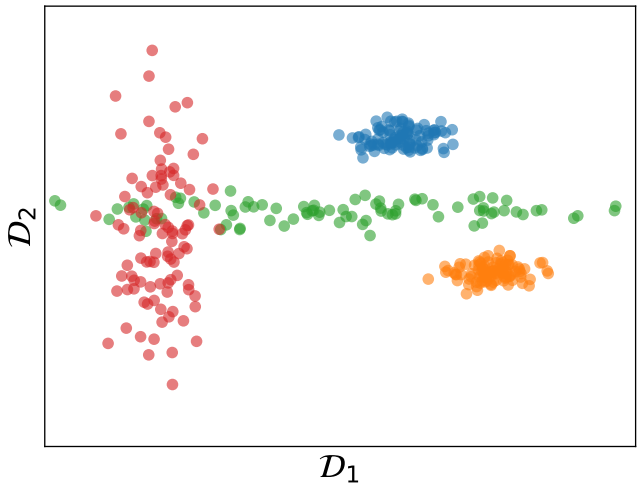
- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

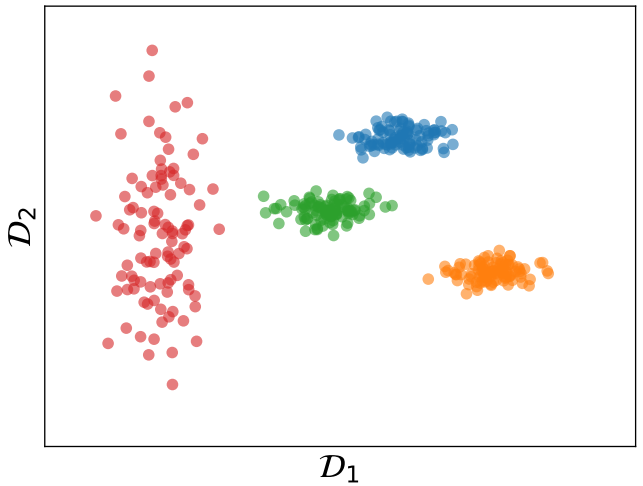


Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

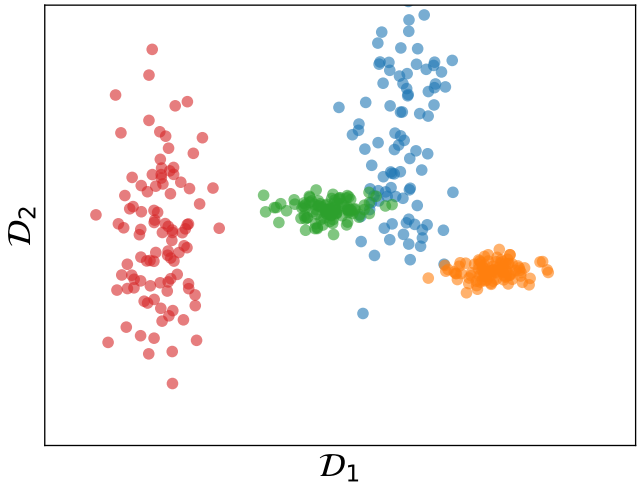


Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

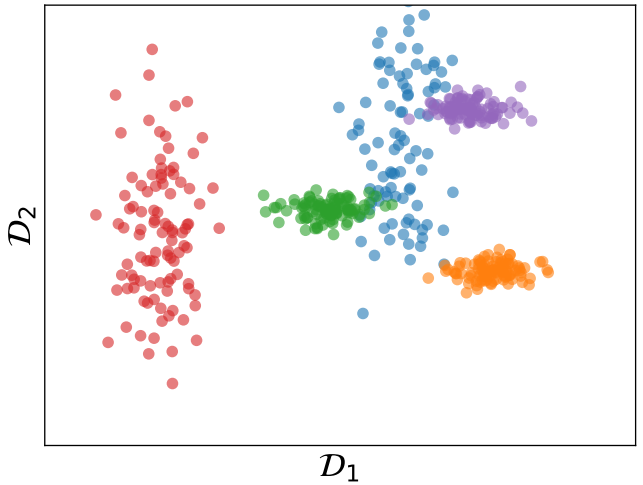


Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

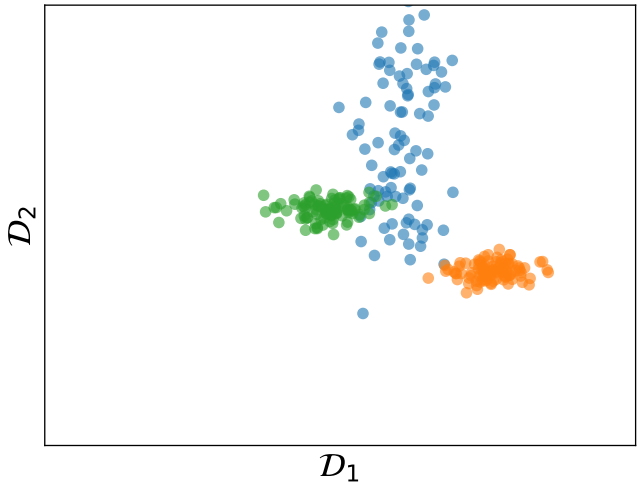


Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

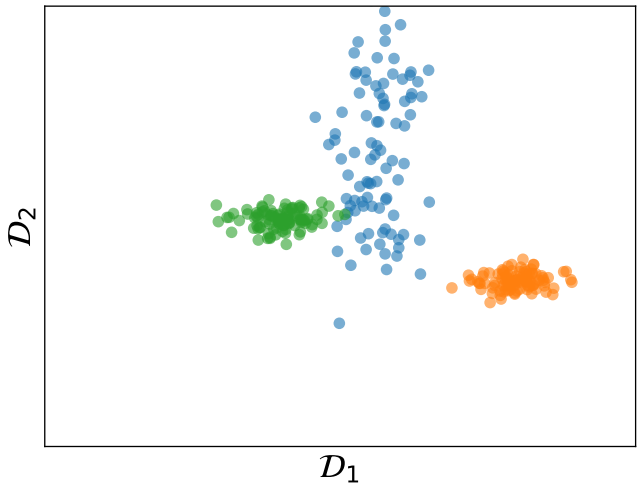


Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.

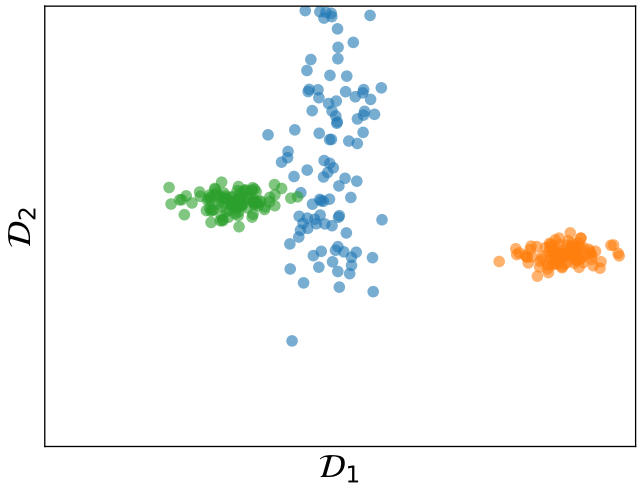


Data streams e.g. [Sim et al. DMKD 2013]

- Data **objects continuously arriving** over time.
- Main **subspace clustering** families **extended** to **data streams**.

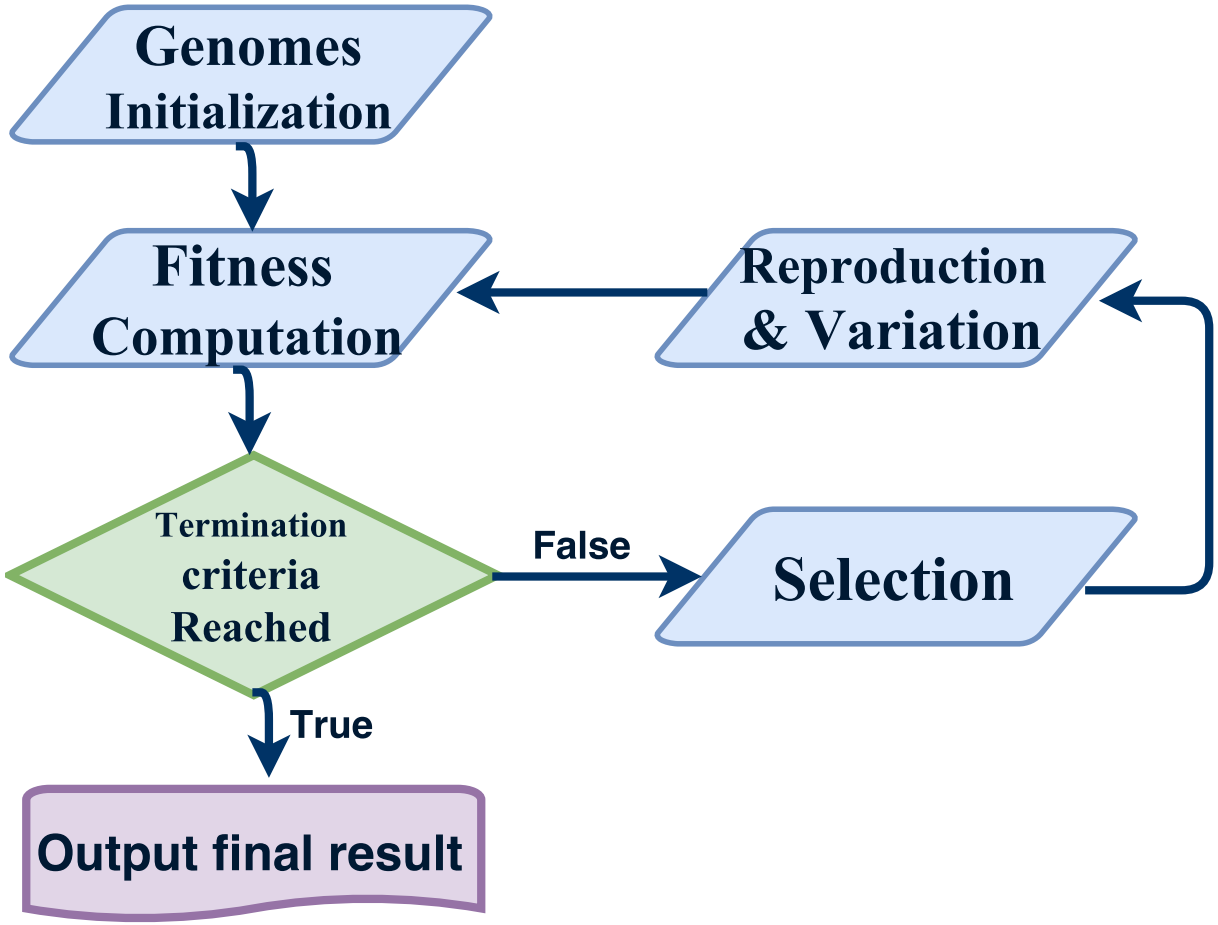
Challenges

- **Single pass** algorithms.
- **On-the-fly** clustering.
- **Dynamic changes** in the data stream.



Outline

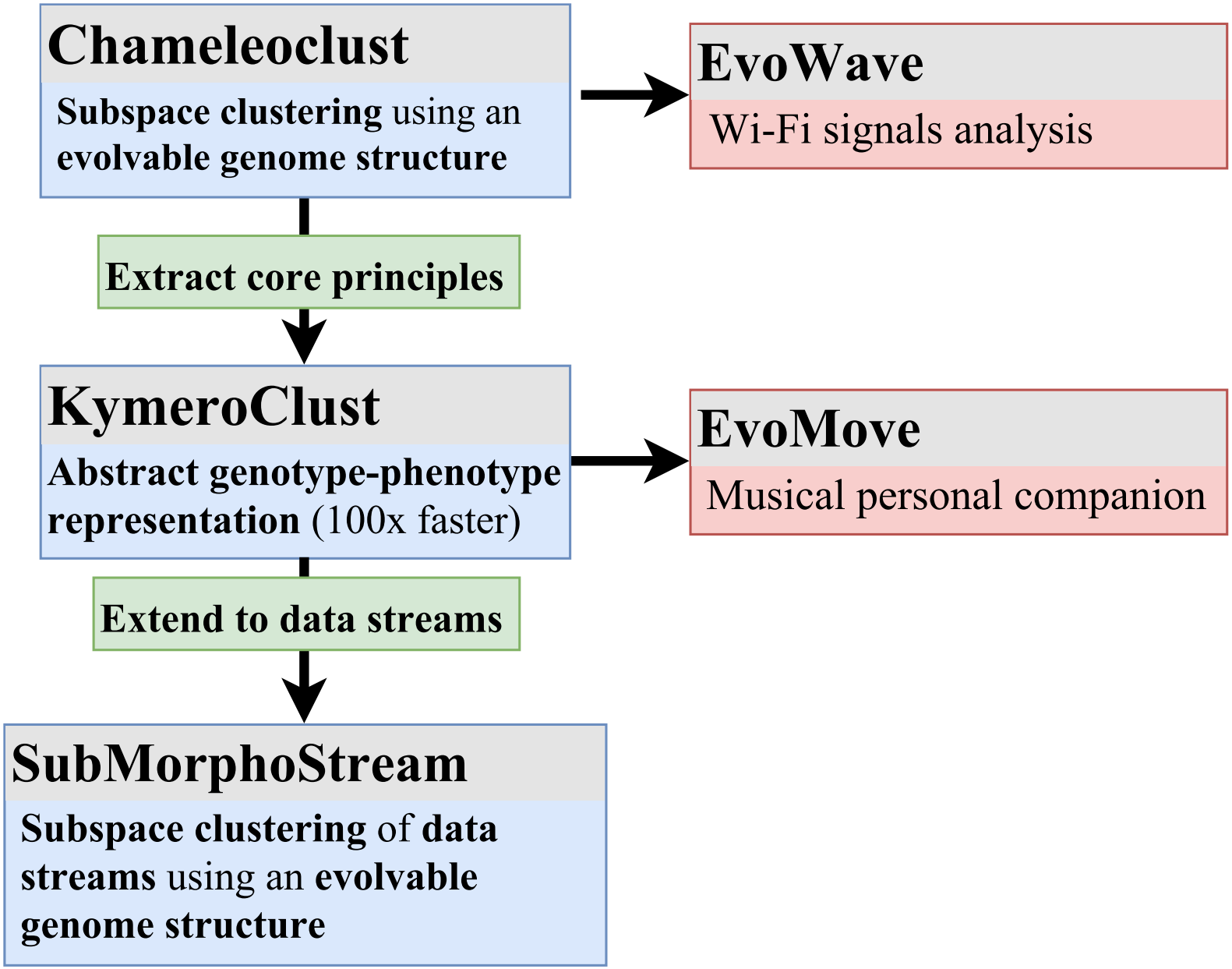
Evolutionary Algorithms general scheme



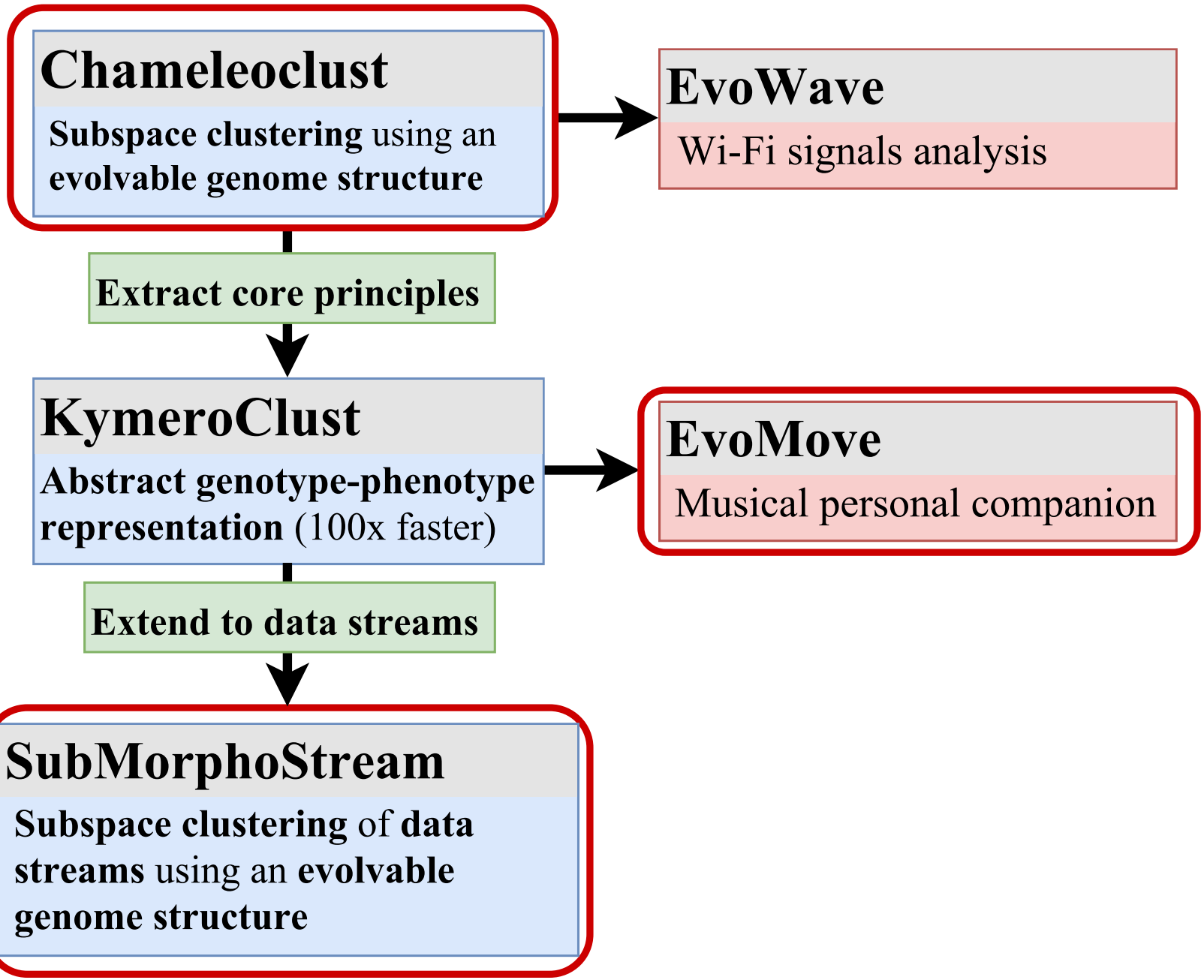
Evolutionary Algorithms applied to clustering

- **Several** evolutionary algorithms for **traditional clustering** reviewed in [Hruschka et al. IEEE TSMC 2009].
- **Few** evolutionary algorithm for **subspace clustering** (incorporating non **non-evolutionary** stages).
  [Sarafis et al. 2007][Vahdat et al. 2013]
- **Few** evolutionary algorithms for **clustering of data streams** [León et al. 2010][Veloza et al. 2013].
- **No** evolutionary algorithms for **subspace clustering of data streams**.

Project outline



Project outline



Outline

Chameleoclust

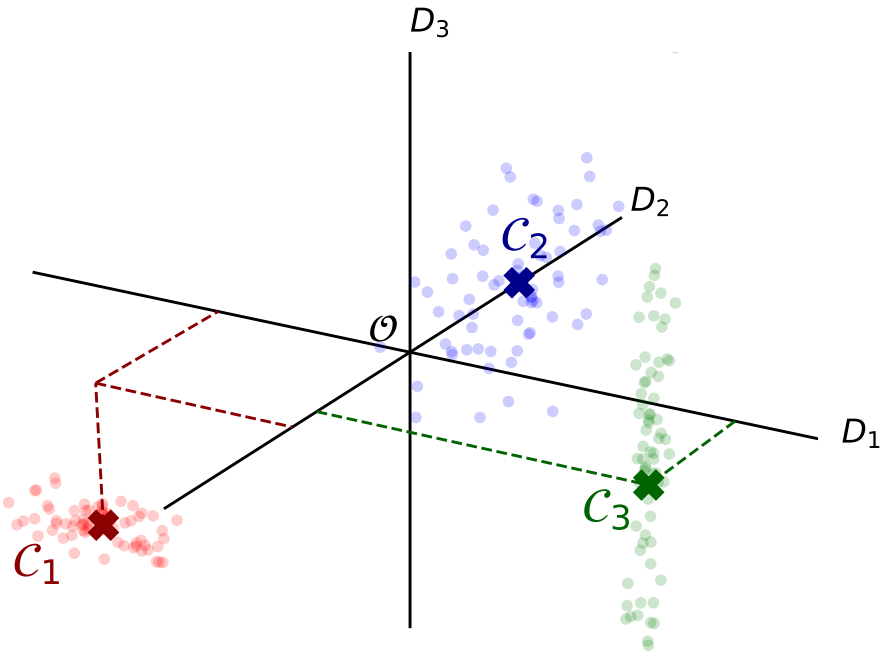
- **Evolvable** genome structure.
  - Variable number of functional and non-functional elements.
  - Flexible organisation of genes.
- Genome structure inspired by **Pearls-on-a-string evolution formalism** [Crombach and Hogeweg, 2007].
- **Bio-inspired** mutations (e.g., large rearrangements).

Chameleoclust

- **Evolvable** genome structure.
  - Variable number of functional and non-functional elements.
  - Flexible organisation of genes.
- Genome structure inspired by **Pearls-on-a-string evolution formalism** [Crombach and Hogeweg, 2007].
- **Bio-inspired** mutations (e.g., large rearrangements).
  - Modify the **genes content** and the **genome structure**.

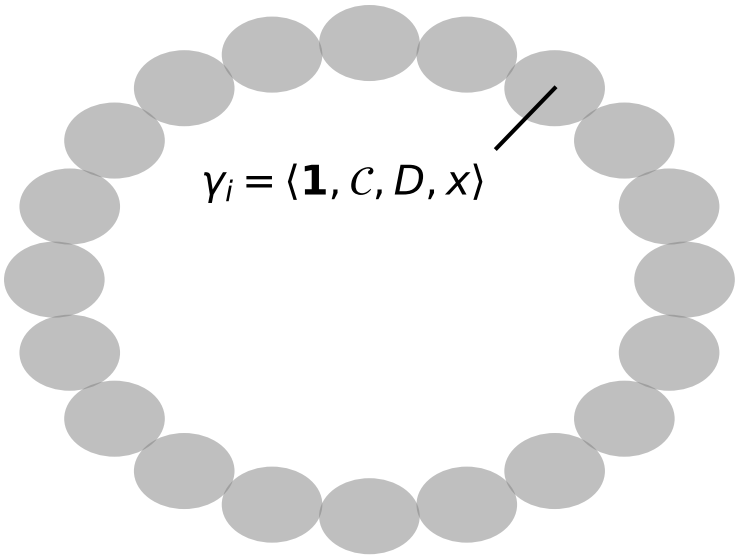
Chameleoclust

- **Evolvable** genome structure.
  - Variable number of functional and non-functional elements.
  - Flexible organisation of genes.
- Genome structure inspired by **Pearls-on-a-string evolution formalism** [Crombach and Hogeweg, 2007].
- **Bio-inspired** mutations (e.g., large rearrangements).
  - Modify the **genes content** and the **genome structure**.
- **Axis-parallel** and **Clustering-Oriented**.

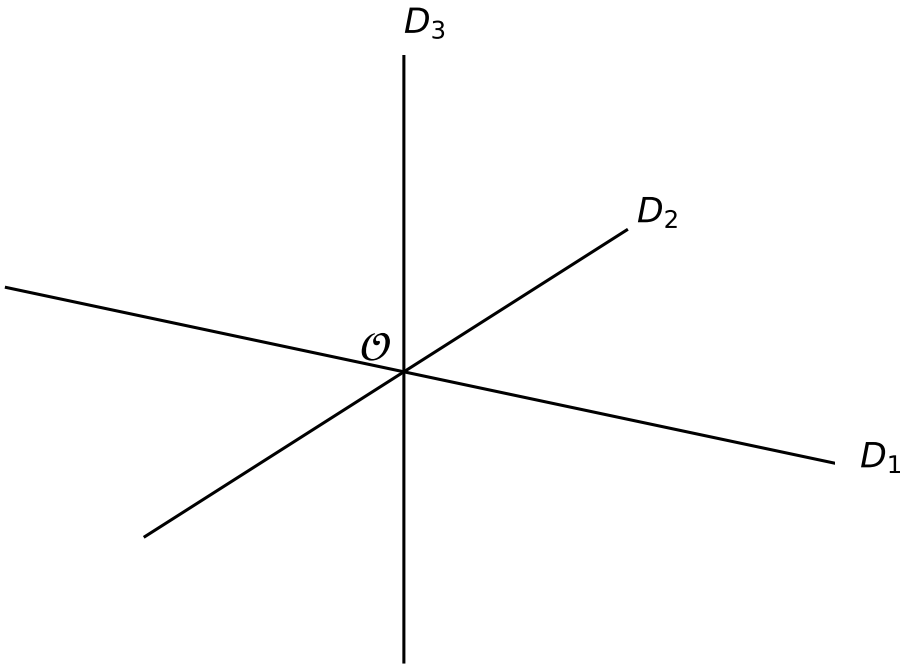


Genotype to Phenotype mapping

Genotype: Γ

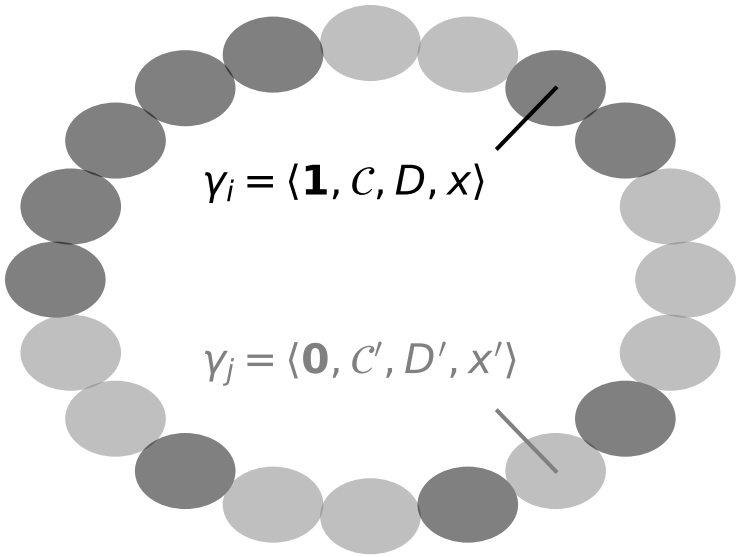


Phenotype: Φ

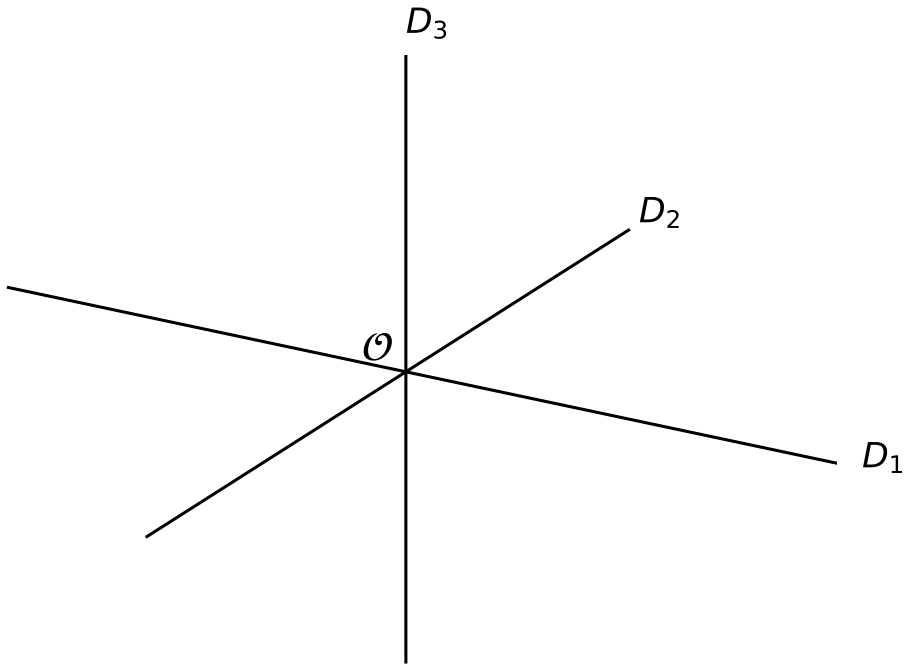


Genotype to Phenotype mapping

Genotype: Γ

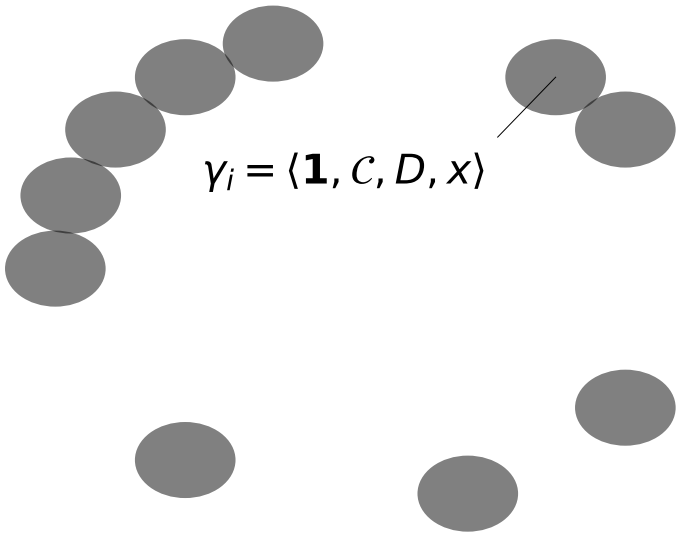


Phenotype: Φ

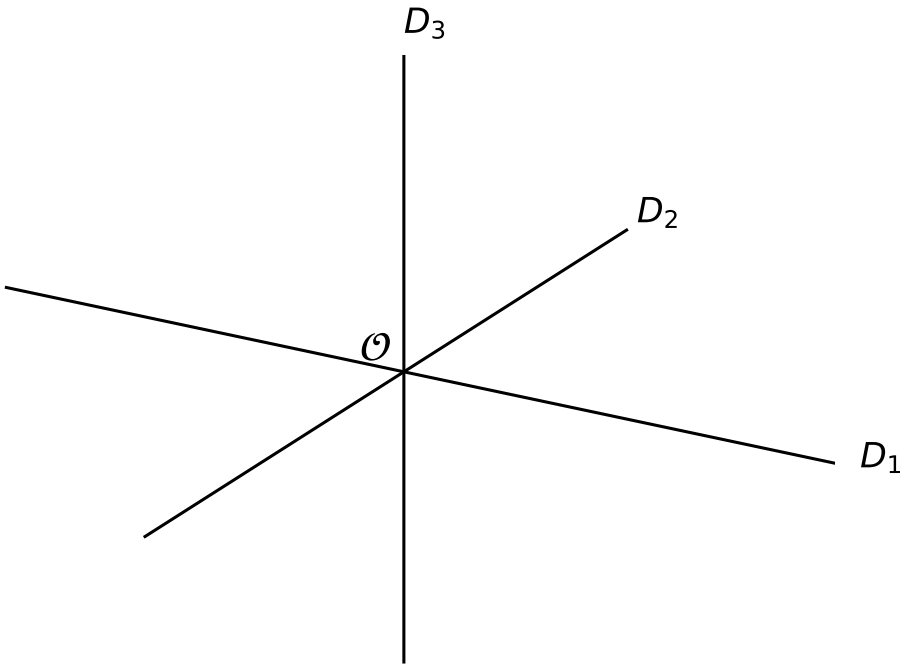


Genotype to Phenotype mapping

Genotype: Γ

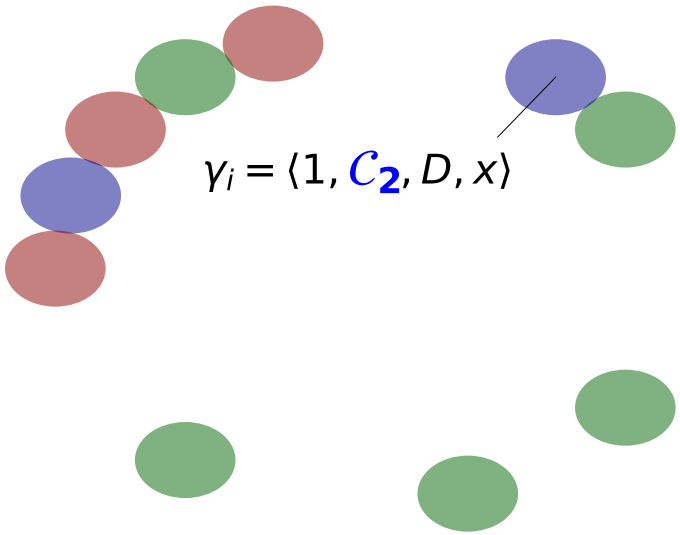


Phenotype: Φ

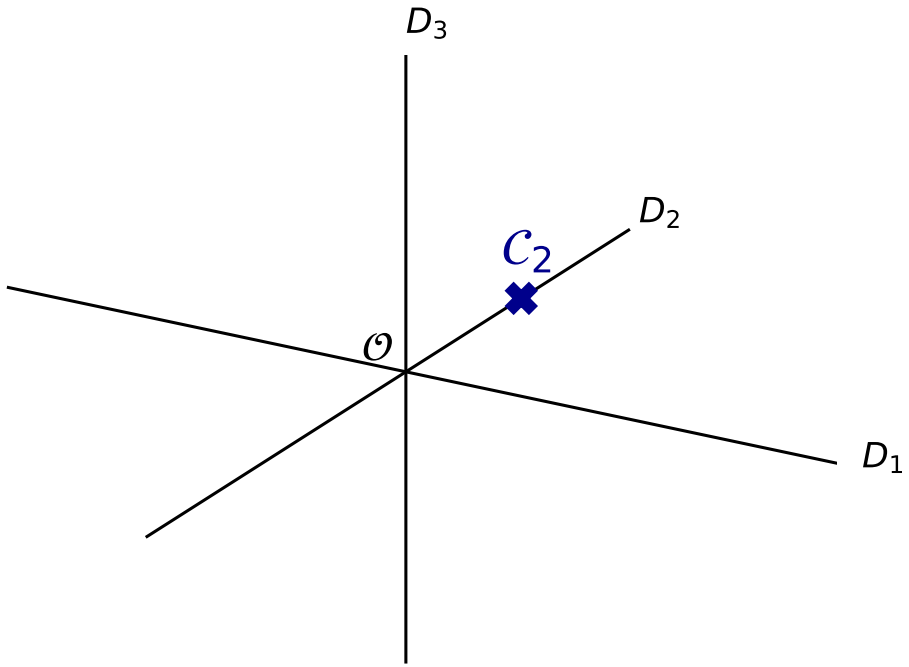


Genotype to Phenotype mapping

Genotype: Γ

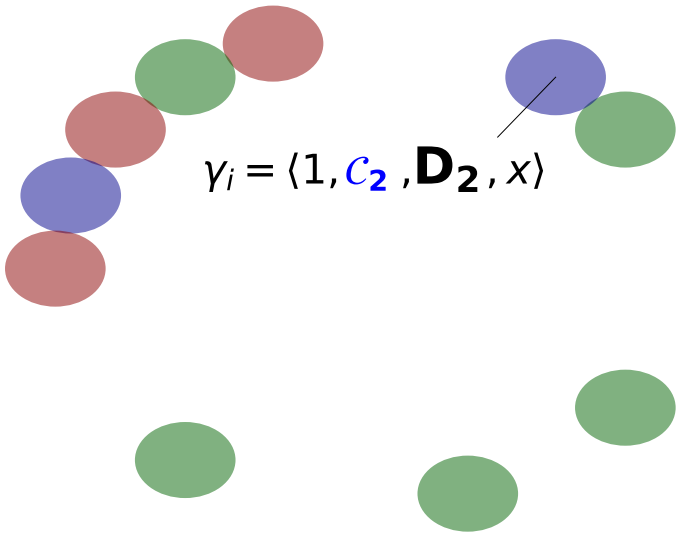


Phenotype: Φ

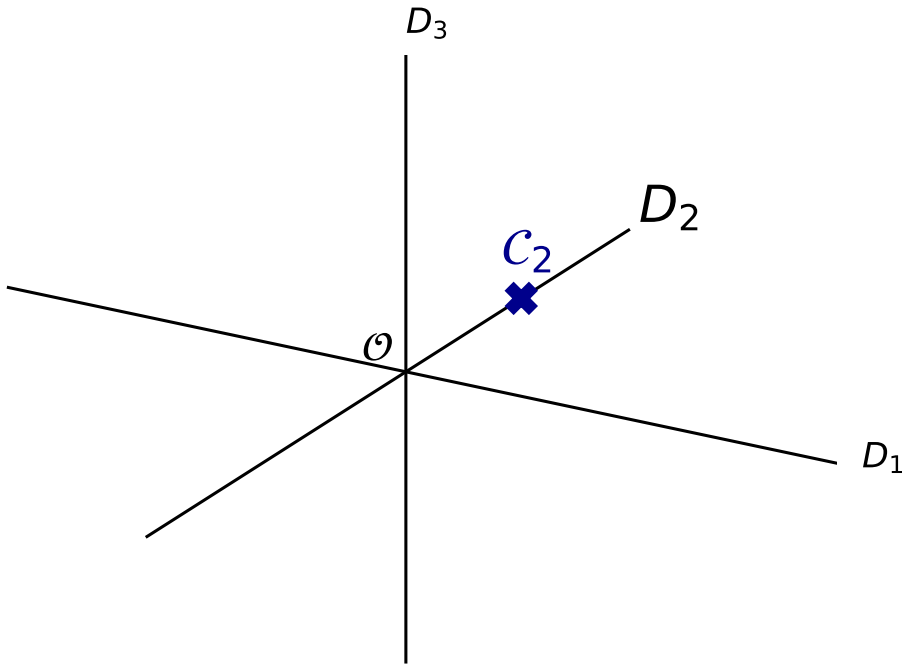


Genotype to Phenotype mapping

Genotype: Γ

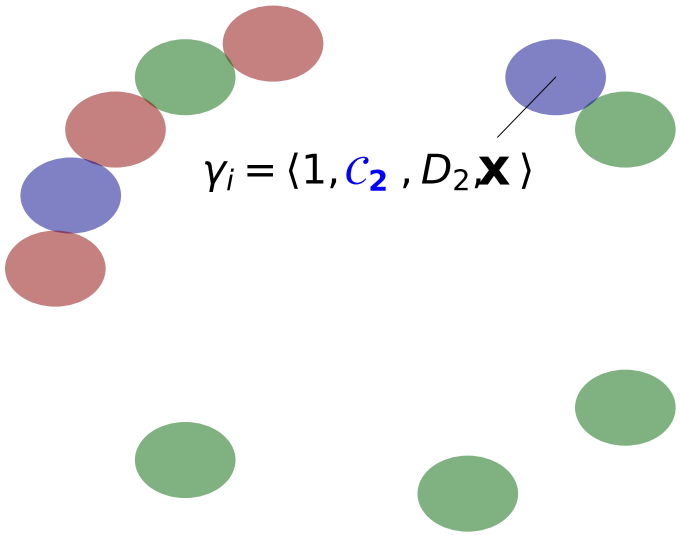


Phenotype: Φ

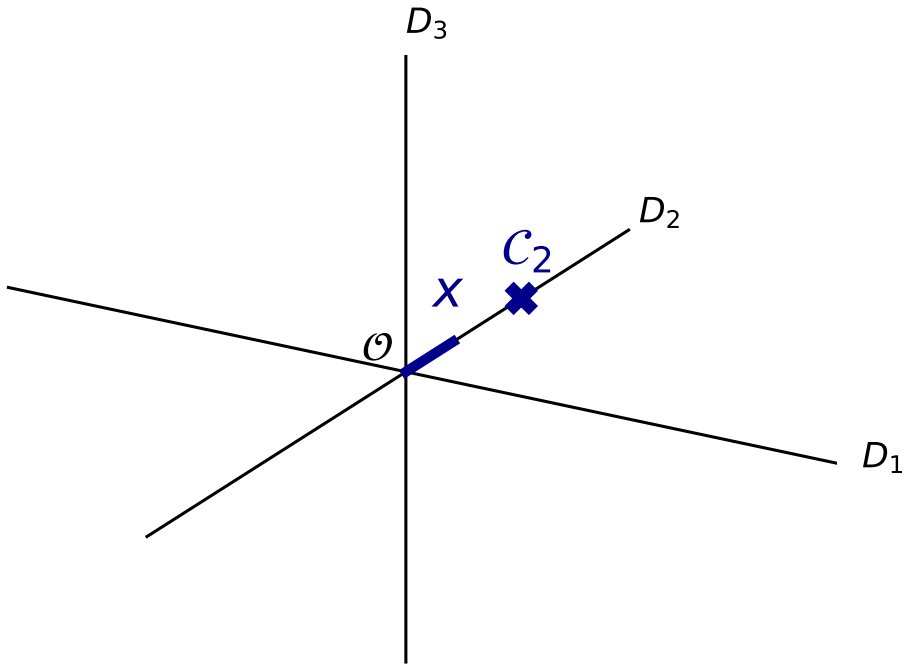


Genotype to Phenotype mapping

Genotype: Γ

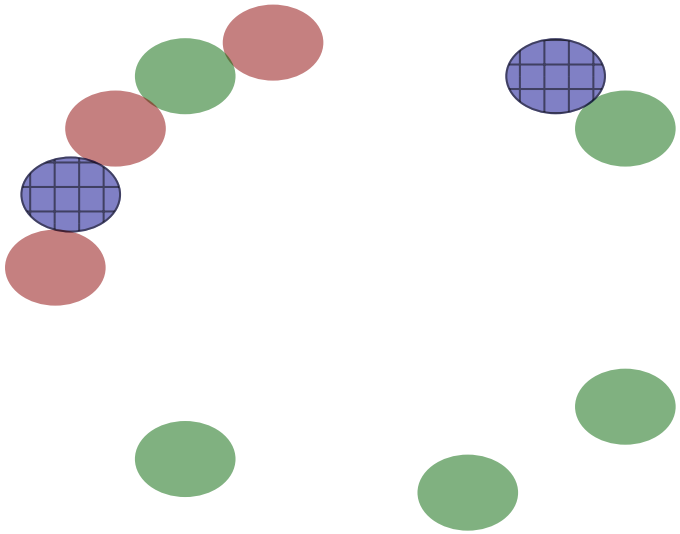


Phenotype: Φ

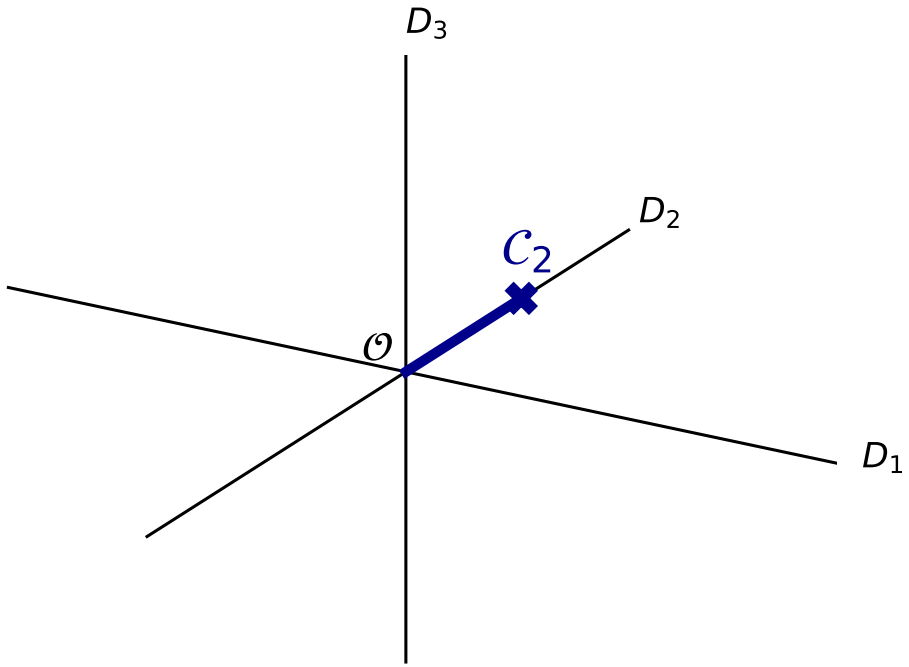


Genotype to Phenotype mapping

Genotype: Γ

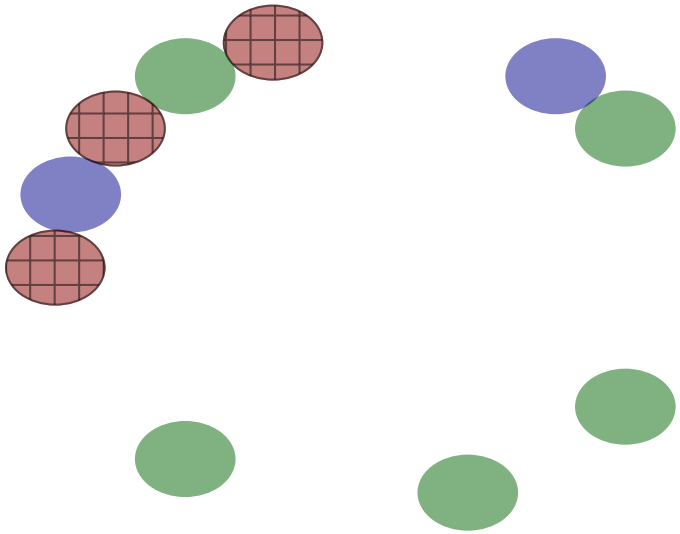


Phenotype: Φ

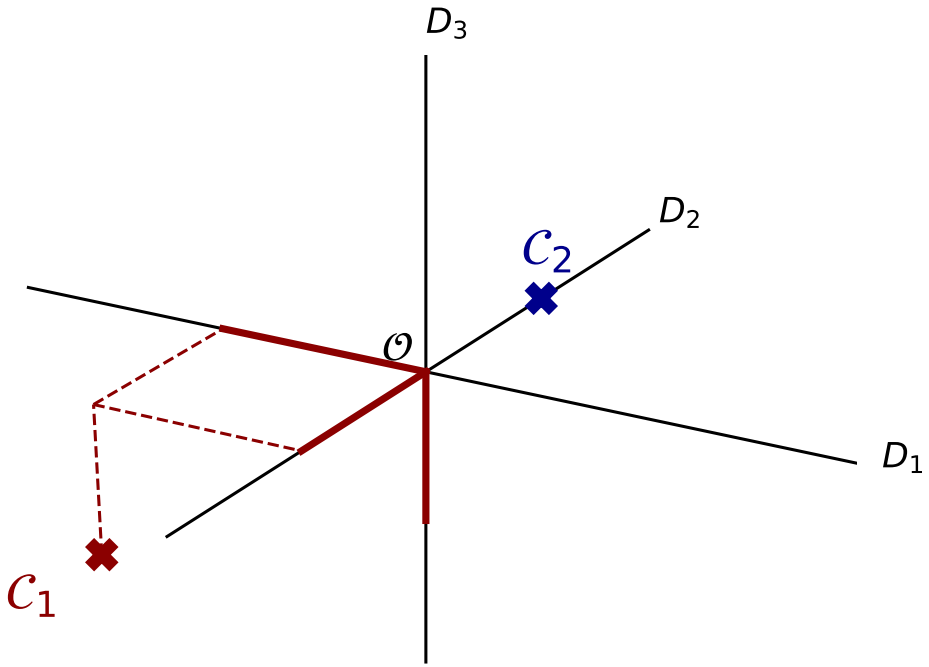


Genotype to Phenotype mapping

Genotype: Γ

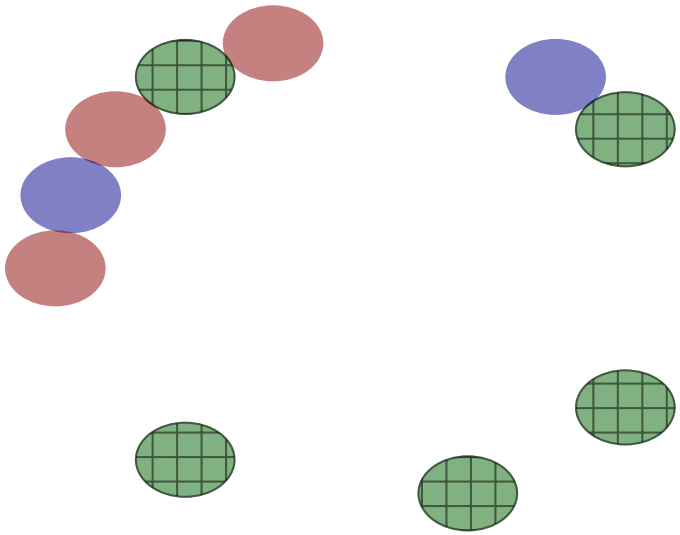


Phenotype: Φ

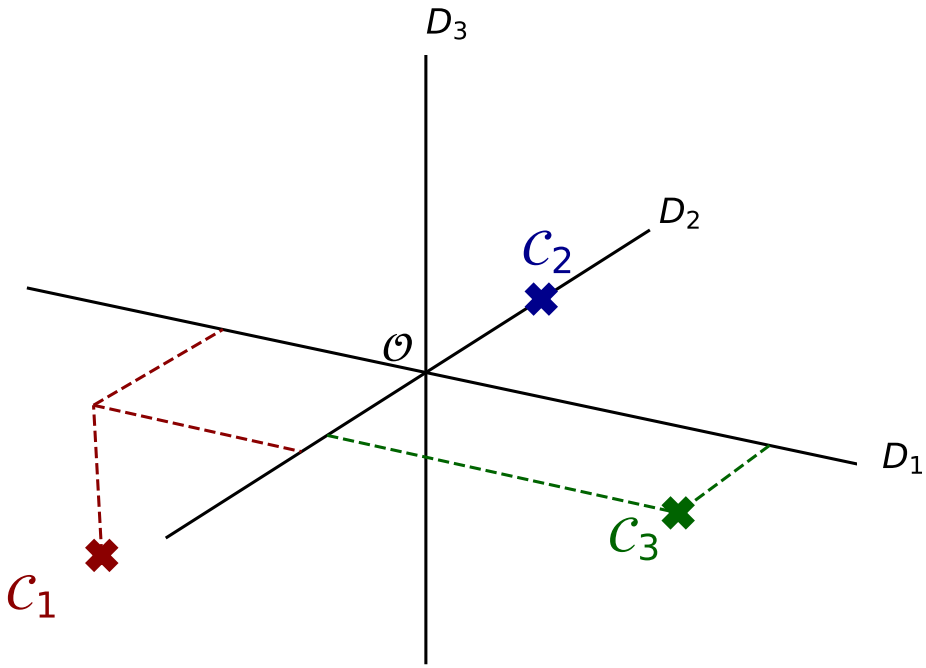


Genotype to Phenotype mapping

Genotype: Γ



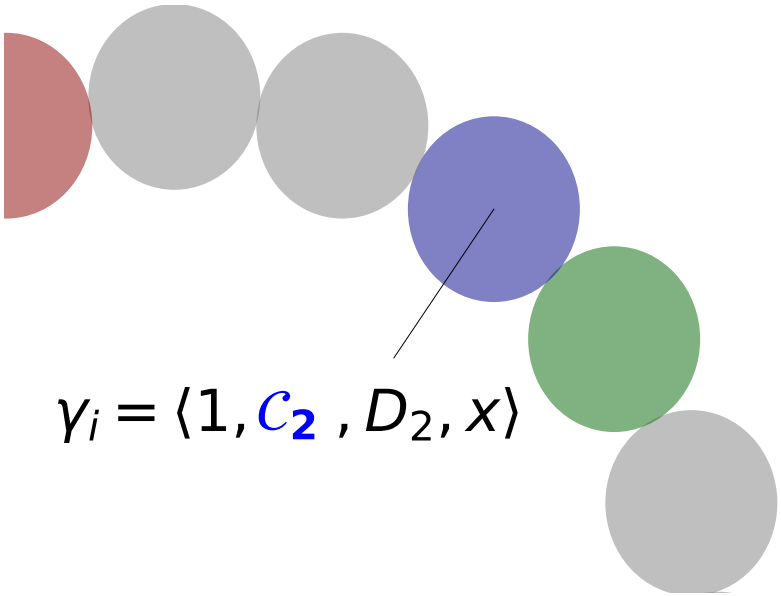
Phenotype: Φ



This **genome structure** allows to encode **different number of clusters** described in their **own subspaces**.

Mutational operators
Point mutations

Before mutation

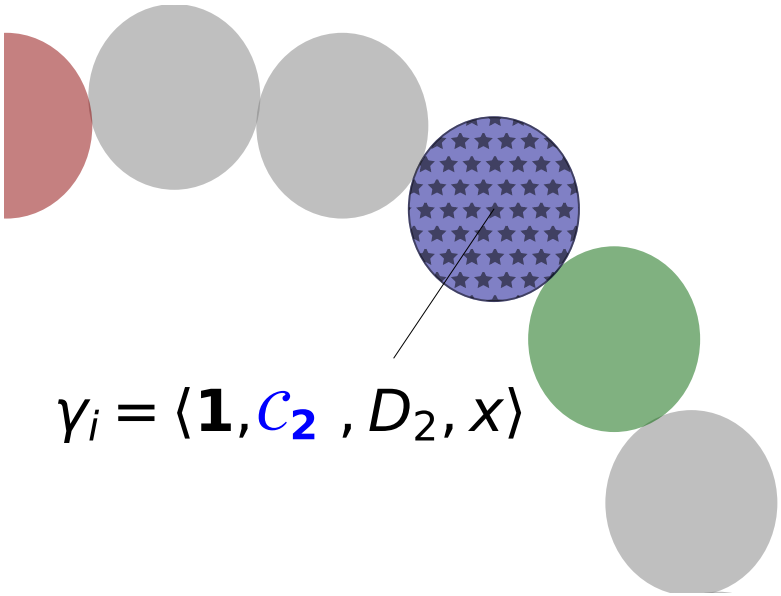


$\gamma_i = \langle 1, \mathcal{C}_{\mathbf{2}}, D_2, x \rangle$

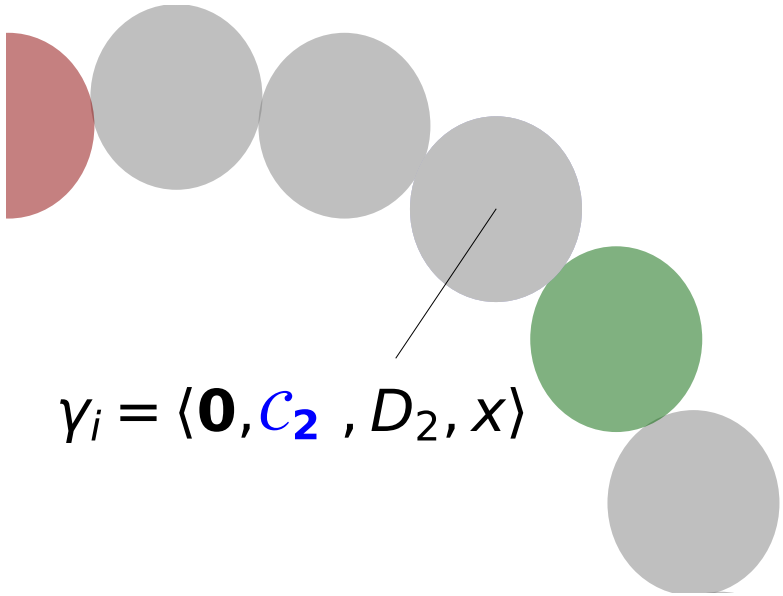
Mutational operators
Point mutations

- **Functional** $\leftrightarrow$ **non-functional** (g).

Before mutation



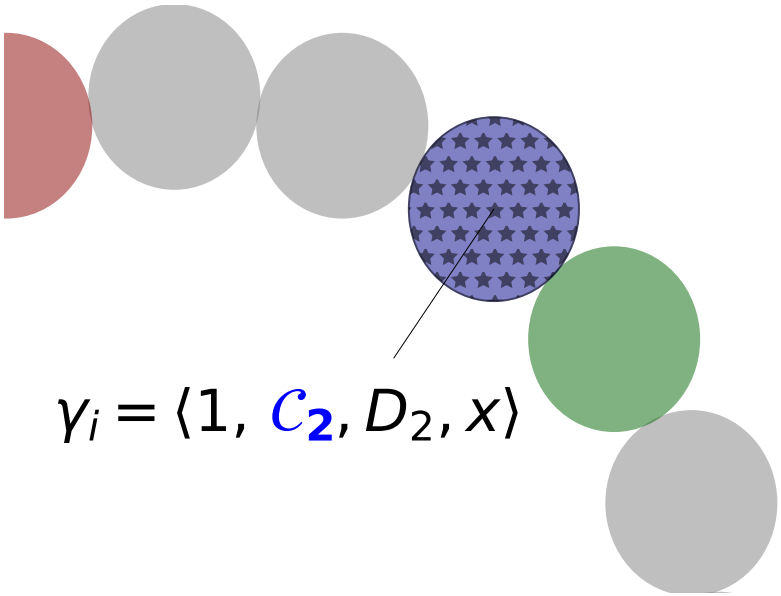
After mutation



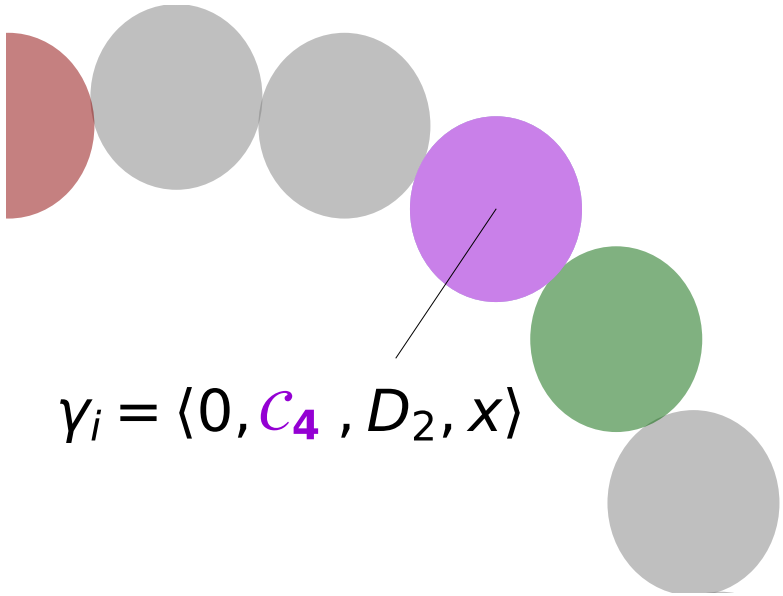
Mutational operators
Point mutations

- **Functional** $\leftrightarrow$ **non-functional** (g).
- **Core-point id** $(\mathcal{C})$

Before mutation



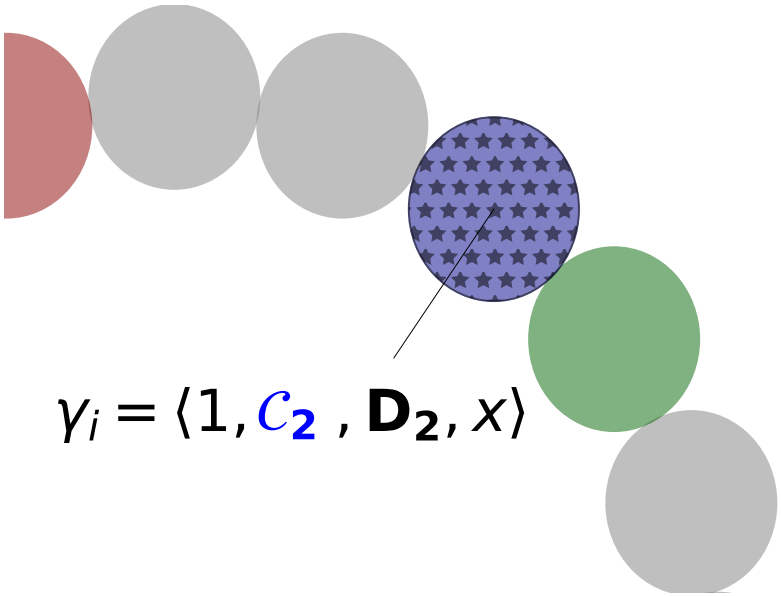
After mutation



Mutational operators
Point mutations

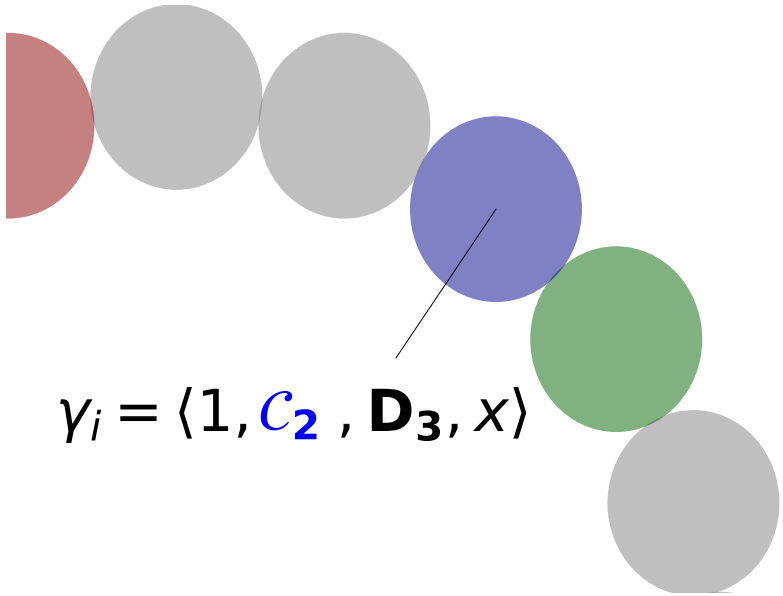
- **Functional** $\leftrightarrow$ **non-functional** (g).
- **Core-point id** $(\mathcal{C})$
- **Dimension id** $(\mathcal{D})$

Before mutation



$\gamma_i = \langle 1, \mathcal{C}_{\mathbf{2}}, \mathbf{D_2}, x \rangle$

After mutation

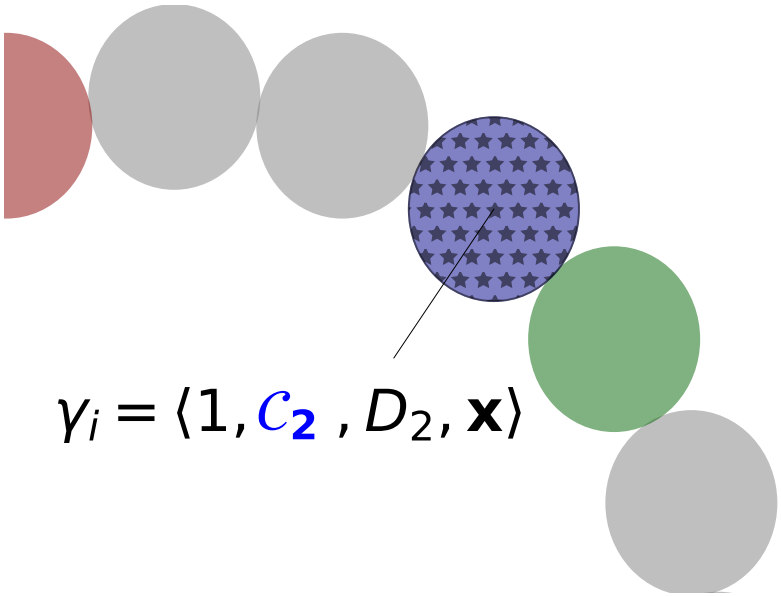


$\gamma_i = \langle 1, \mathcal{C}_{\mathbf{2}}, \mathbf{D_3}, x \rangle$

Mutational operators
Point mutations

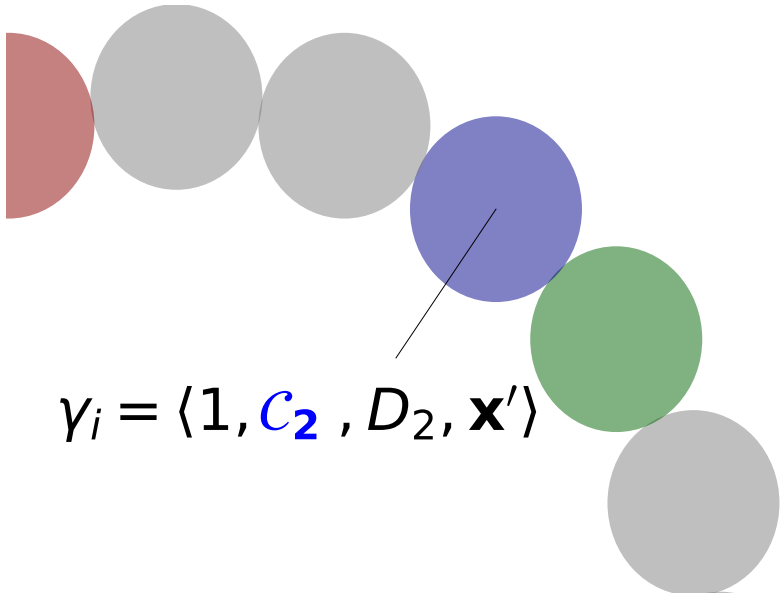
- **Functional** $\leftrightarrow$ **non-functional** (g).
- **Core-point id** $(\mathcal{C})$
- **Dimension id** $(\mathcal{D})$
- **Contribution** (x)

Before mutation



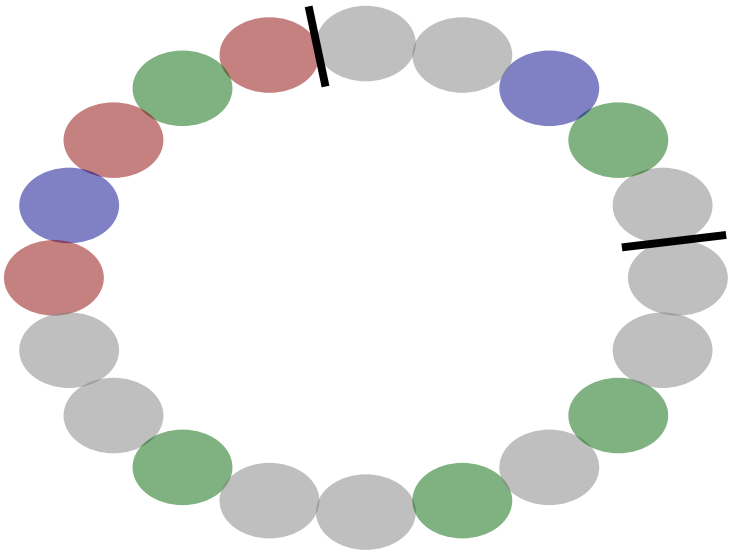
$\gamma_i = \langle 1, \mathcal{C}_2, D_2, \mathbf{x} \rangle$

After mutation



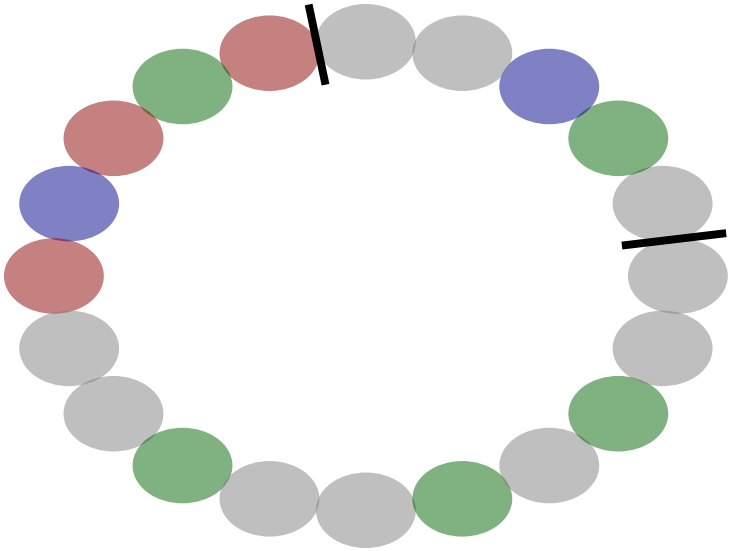
$\gamma_i = \langle 1, \mathcal{C}_2, D_2, \mathbf{x}' \rangle$

Before mutation

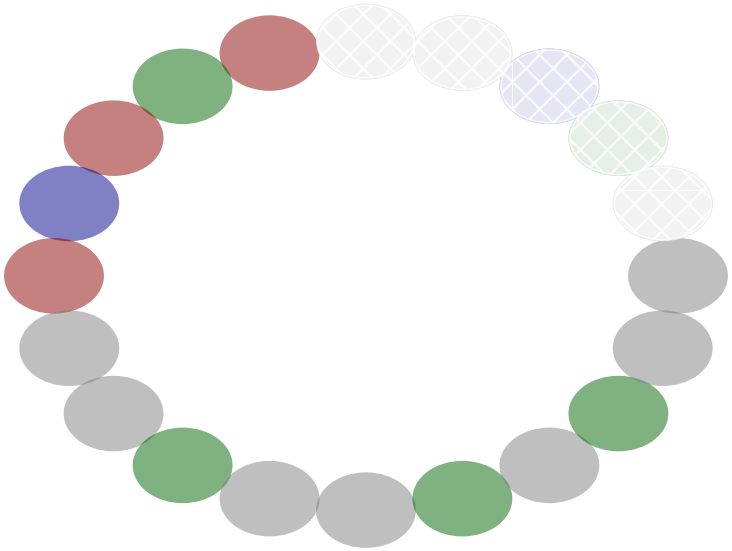


- **Deletion**

Before mutation

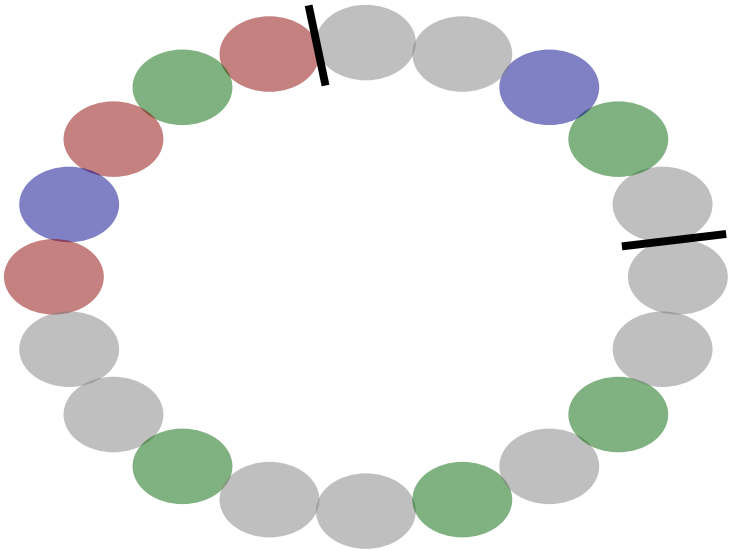


After rearrangement

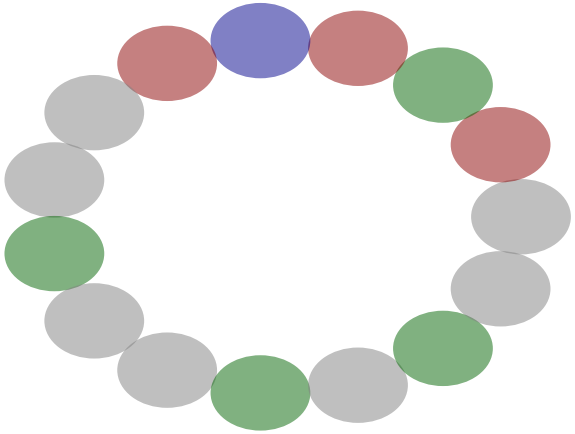


- **Deletion**

Before mutation

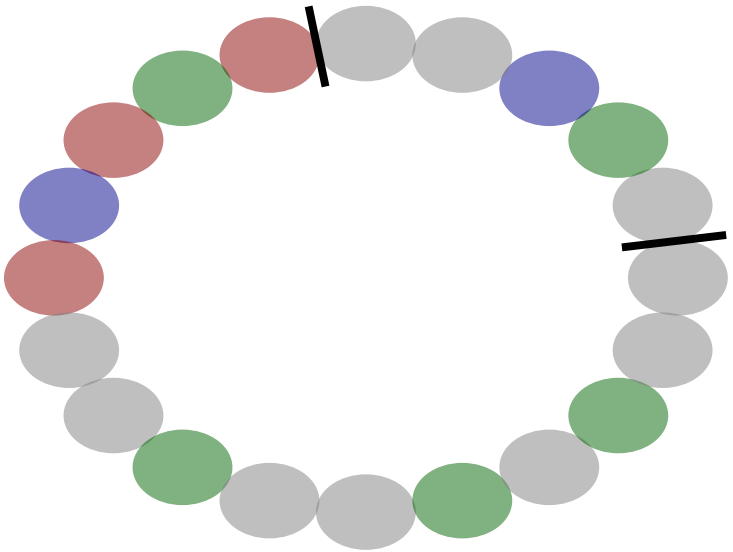


After rearrangement

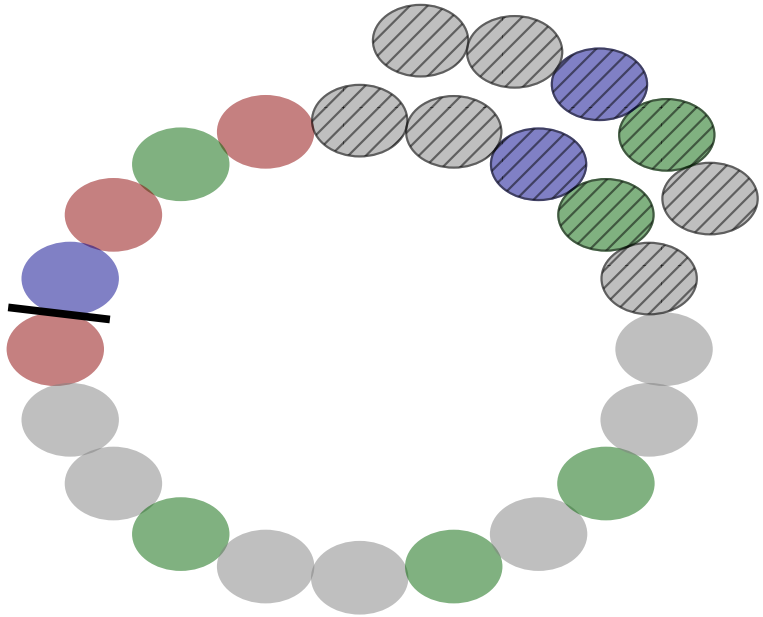


- Deletion
- **Duplication**

Before mutation



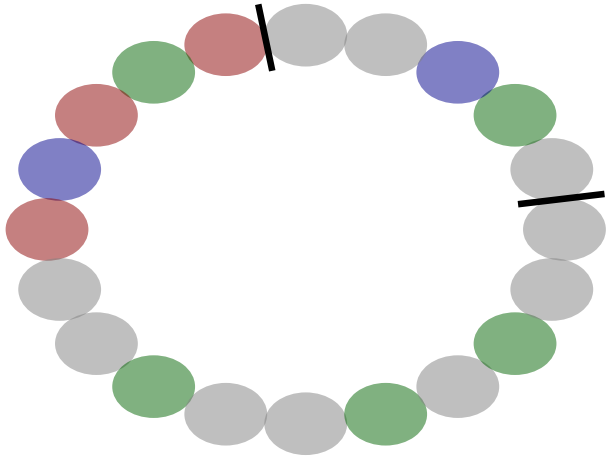
After rearrangement



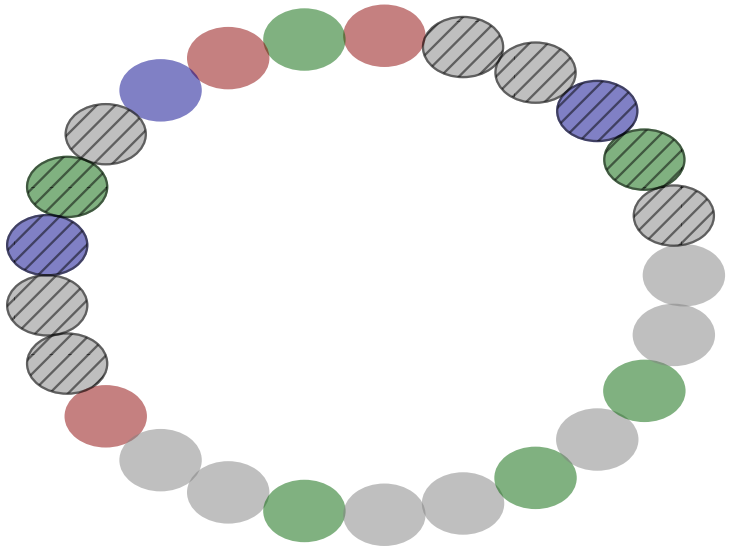
Mutational operators
Large rearrangements: New bio-inspired operators

- Deletion
- **Duplication**

Before mutation

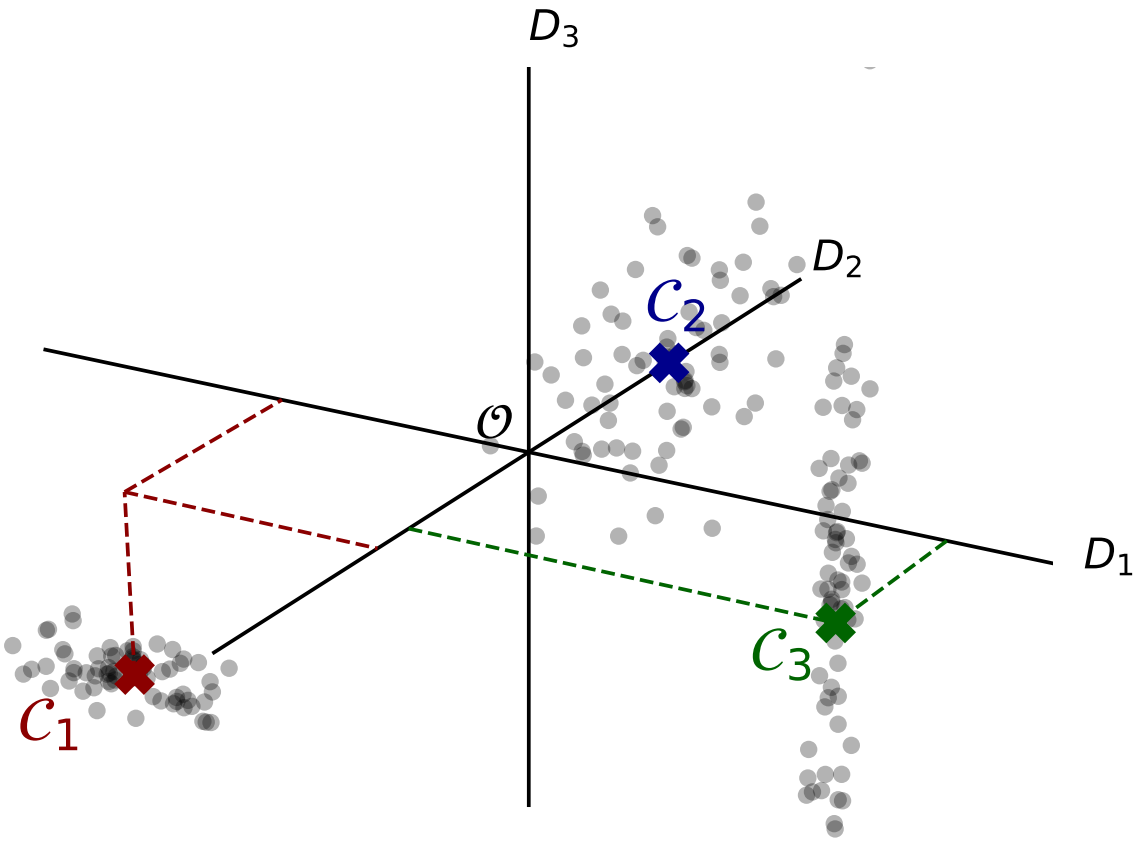


After rearrangement

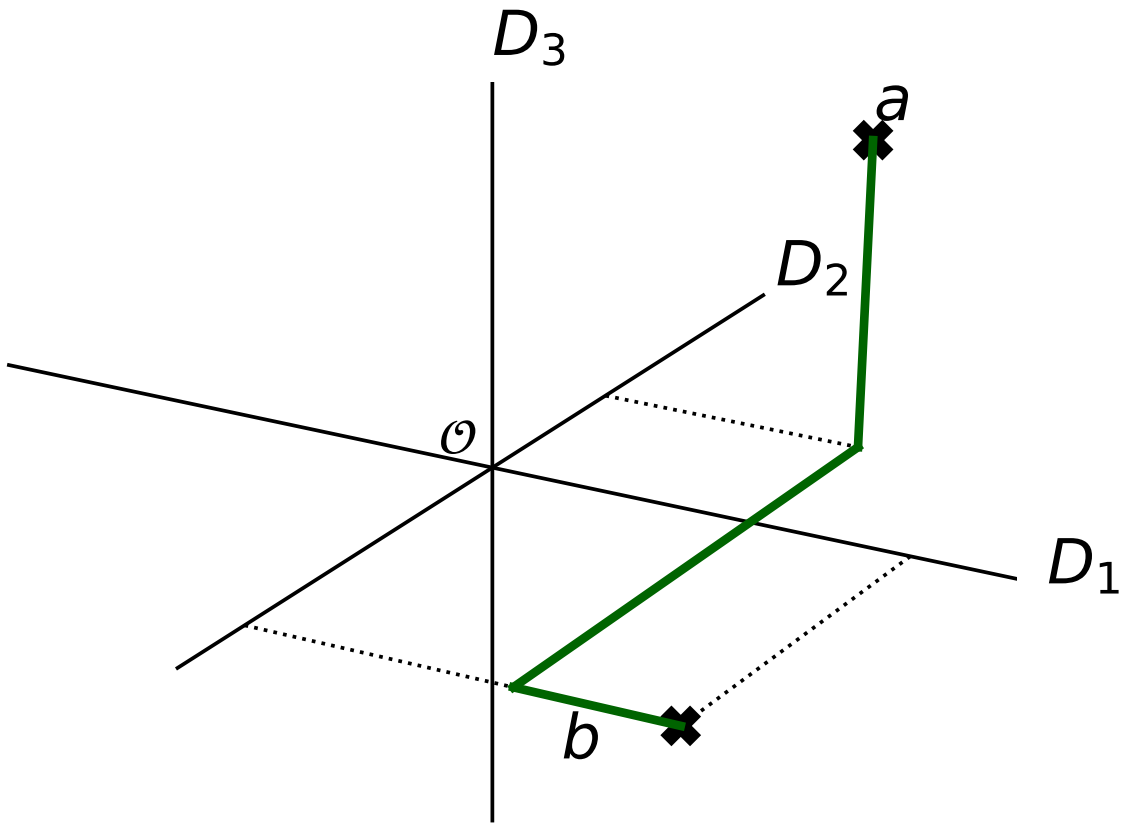


Fitness computation
Assignment Mismatch

- $\mathcal{S}^t$ set of **normalized data objects** observed at **generation** t.
- **Assign objects** in $\mathcal{S}^t$ to **core-points** in phenotype Φ.

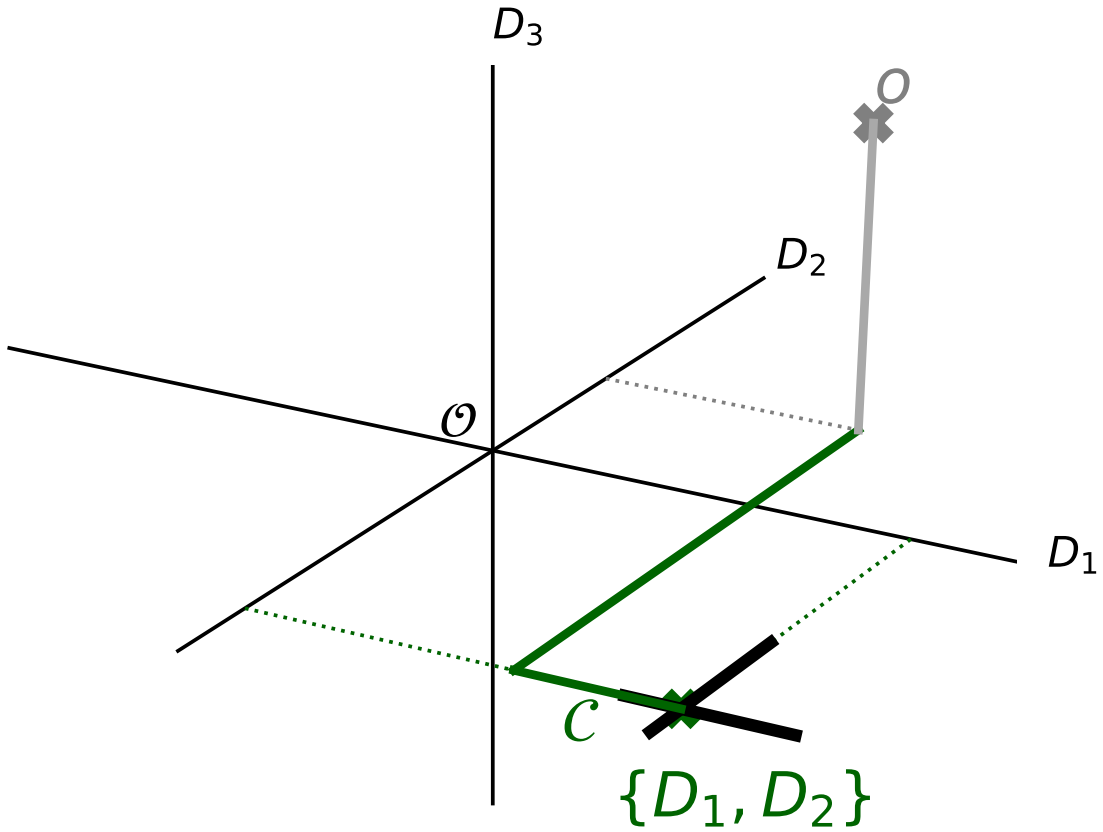


- **Manhattan Segmental Distance** [Aggarwal et al. 1999]



$$d_{\mathcal{D}}(a,b) = \sum_{i \in \mathcal{D}} \frac{|a_i - b_i|}{|\mathcal{D}|}$$

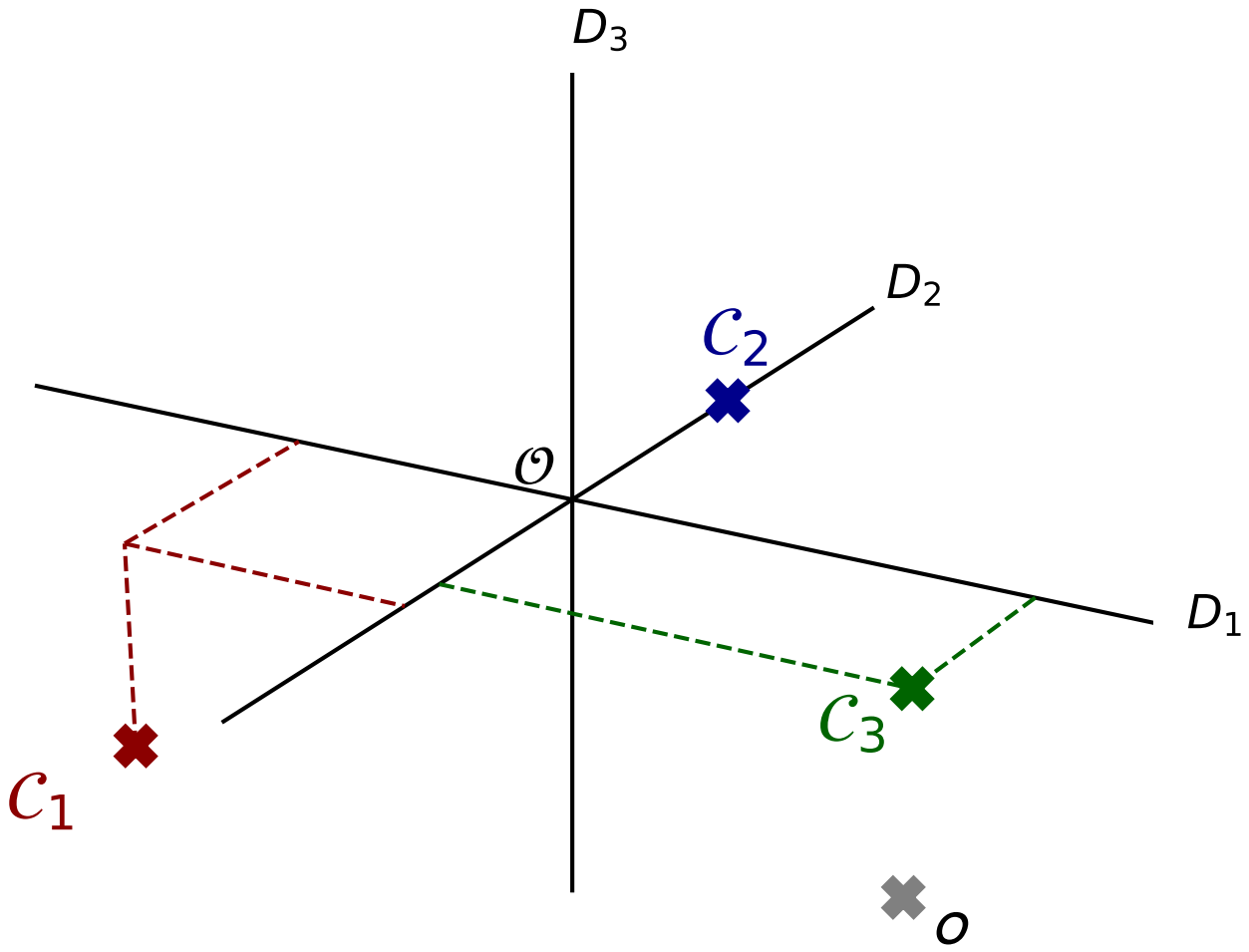
- **Assignment mismatch** $\mathcal{E}(o, \mathcal{C})$



$$\mathcal{E}(o, \mathcal{C}) = \frac{|\mathcal{D}_{\mathcal{C}}| \cdot d_{\mathcal{D}_{\mathcal{C}}}(o, \mathcal{C}) + |\mathcal{D} \setminus \mathcal{D}_{\mathcal{C}}| \cdot d_{\mathcal{D} \setminus \mathcal{D}_{\mathcal{C}}}(o, \mathcal{O})}{|\mathcal{D}|}$$

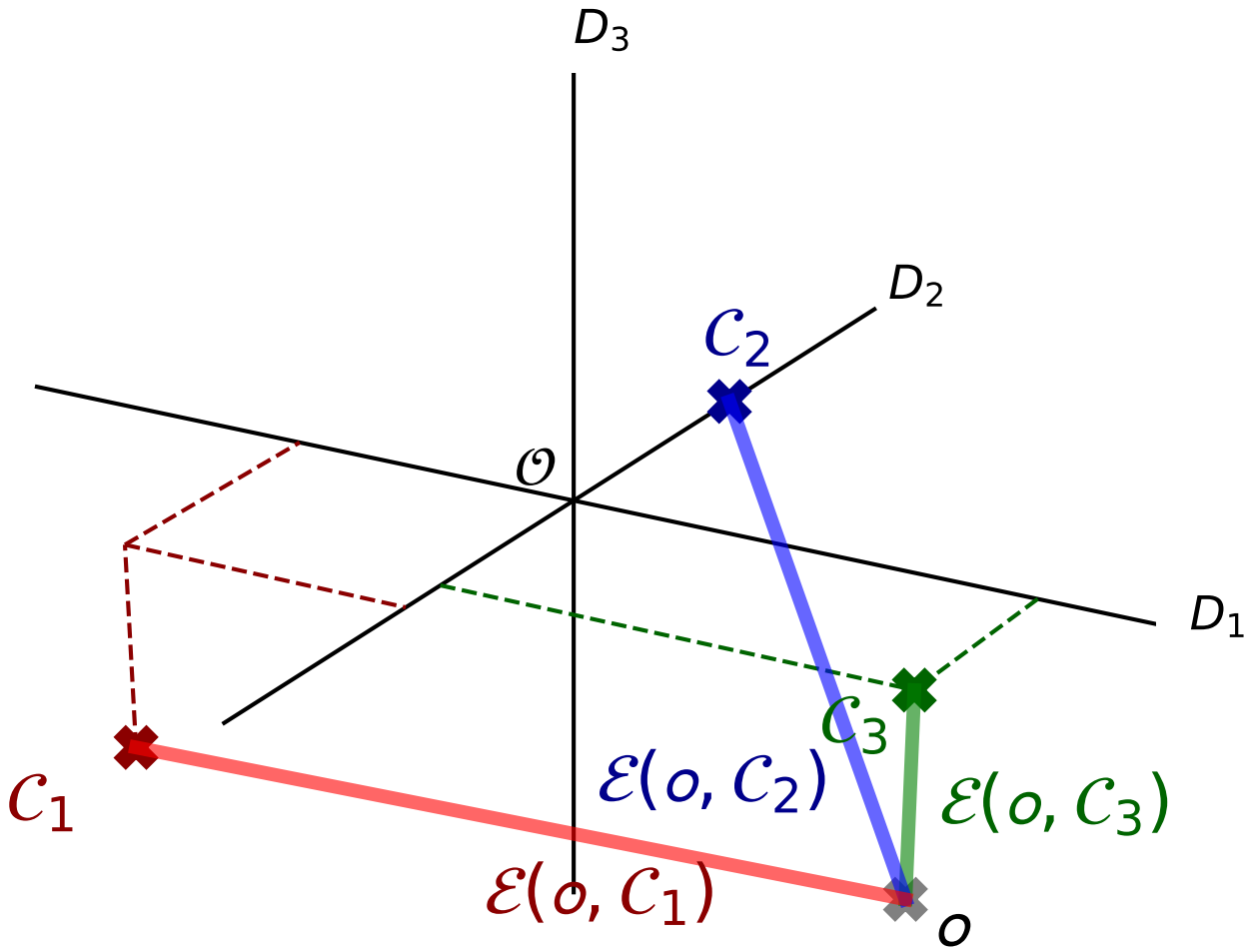
Fitness computation

- **Assign** each **object** $o \in \mathcal{S}^t$ to the closest **core-point** $\mathcal{C} \in \Phi$
- **Assignment mismatch** $\mathcal{E}(o, \mathcal{C}_i)$.



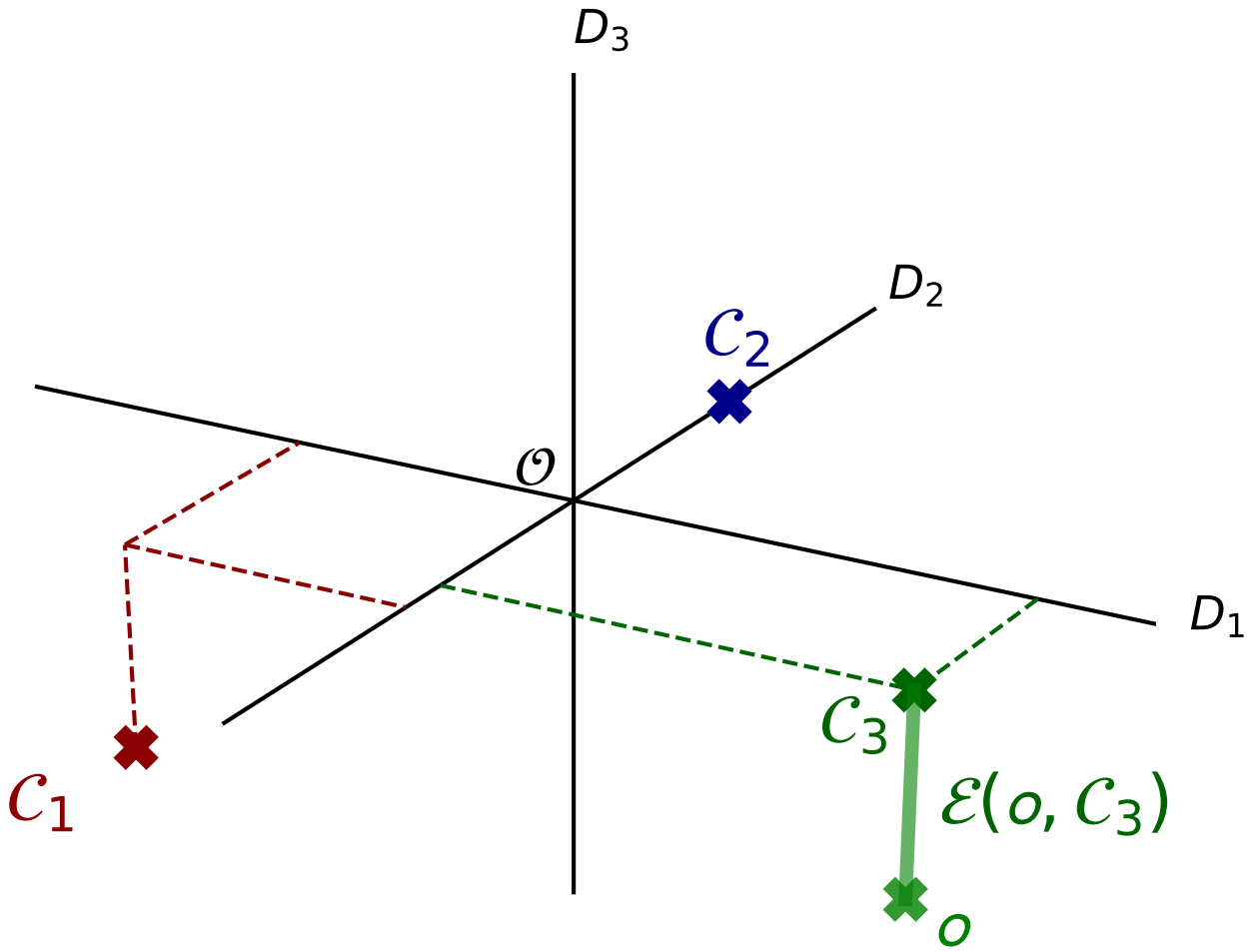
Fitness computation

- **Assign** each **object** $o \in \mathcal{S}^t$ to the closest **core-point** $\mathcal{C} \in \Phi$
- **Assignment mismatch** $\mathcal{E}(o, \mathcal{C}_i)$.



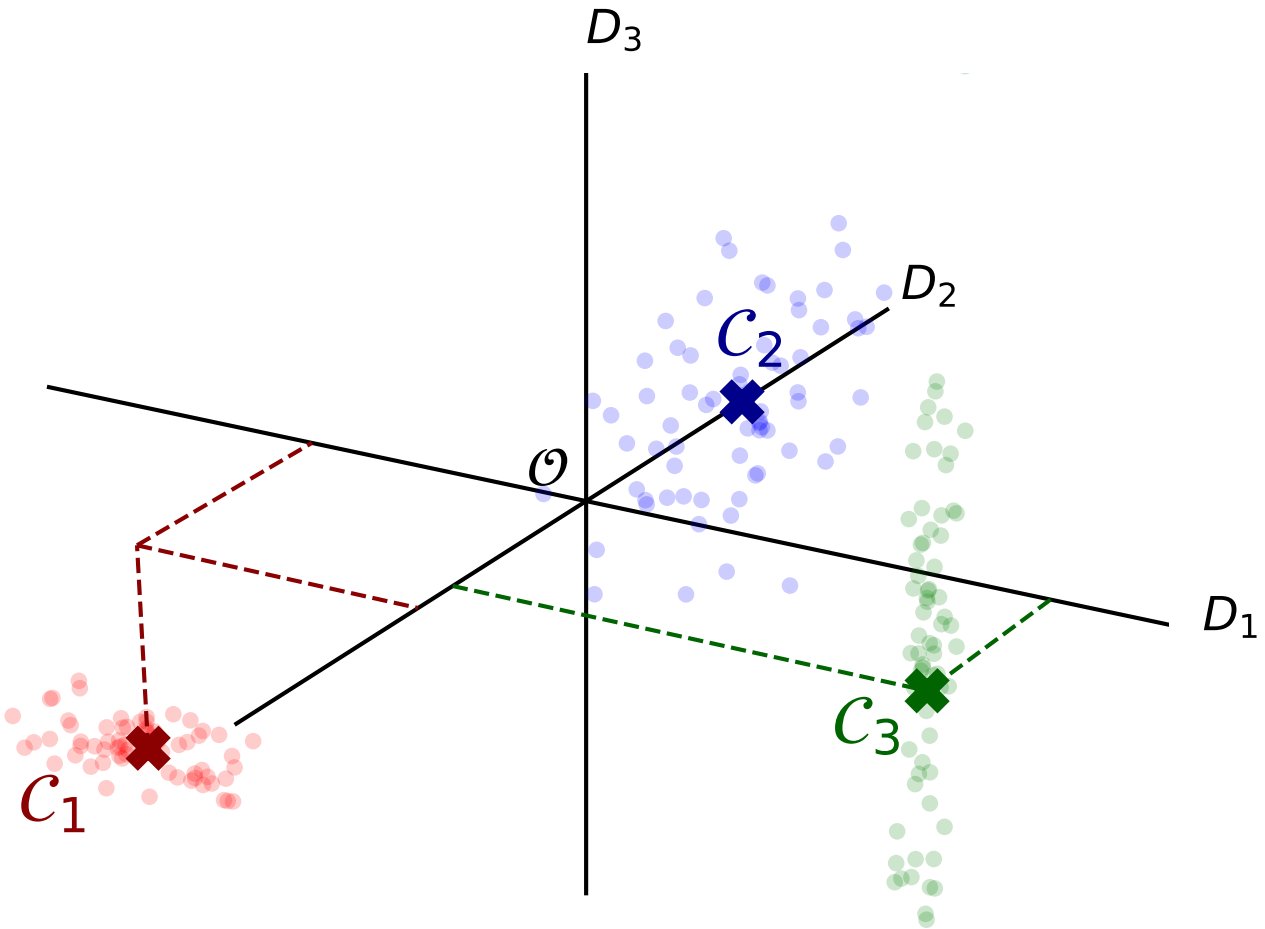
Fitness computation

- **Assign** each **object** $o \in \mathcal{S}^t$ to the closest **core-point** $\mathcal{C} \in \Phi$
- **Assignment mismatch** $\mathcal{E}(o, \mathcal{C}_i)$.



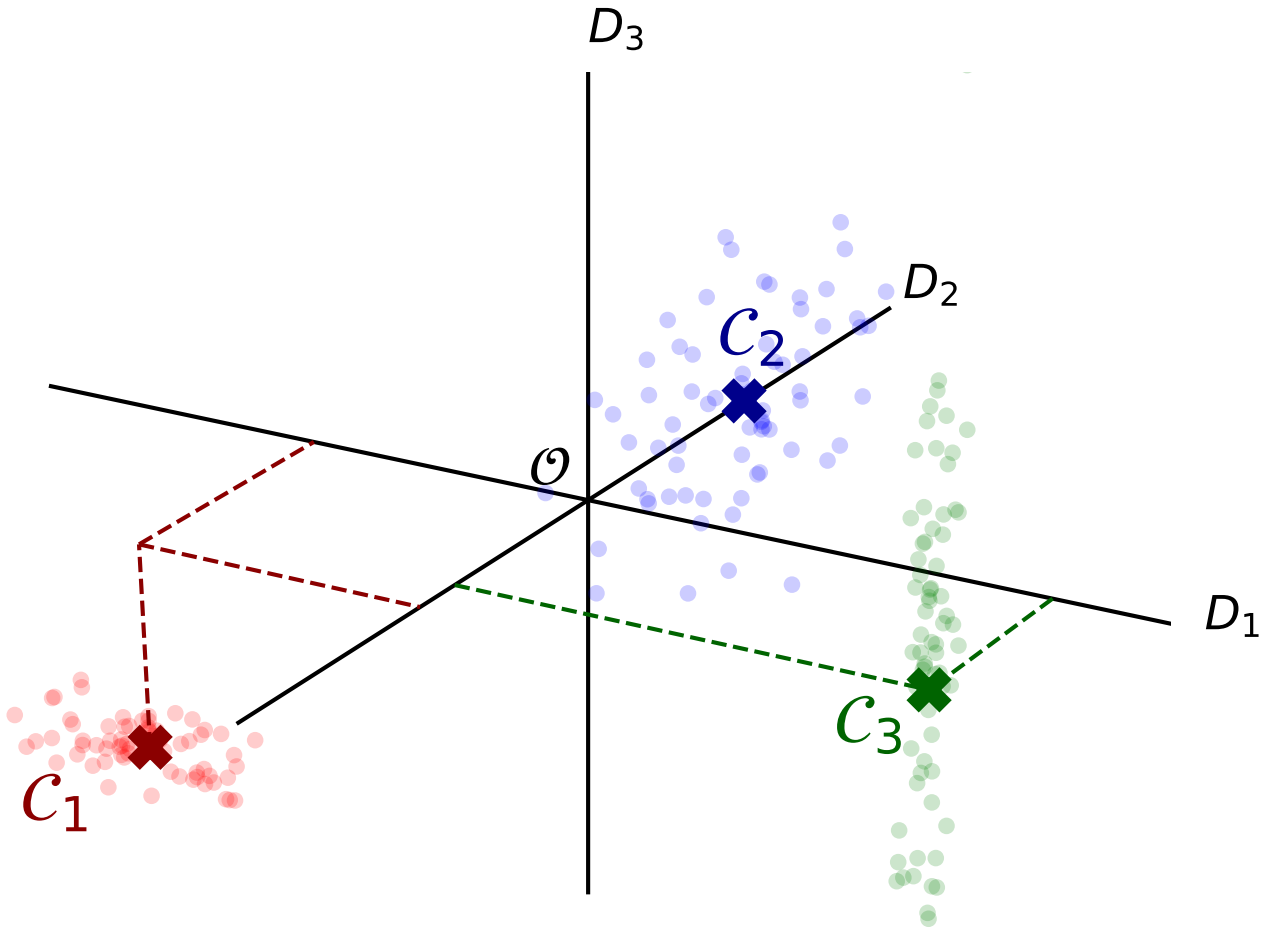
Fitness computation

- **Assign** each **object** $o \in \mathcal{S}^t$ to the closest **core-point** $\mathcal{C} \in \Phi$
- **Assignment mismatch** $\mathcal{E}(o, \mathcal{C}_i)$.



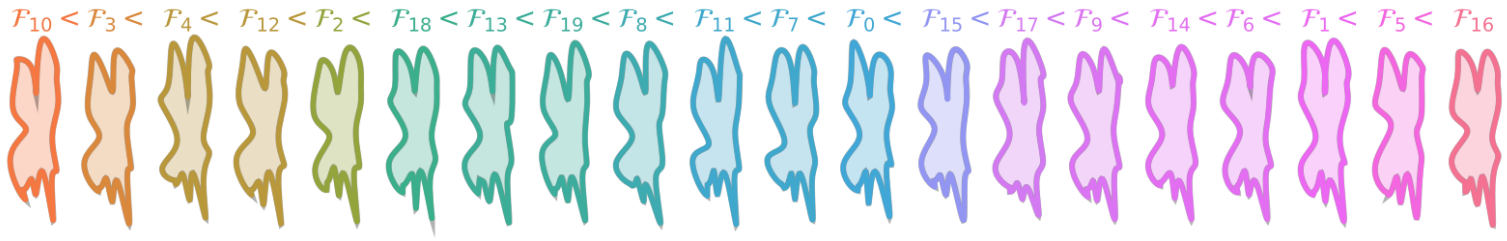
Fitness computation

$$\mathcal{F}(\mathcal{S}^t, \Phi) = -\frac{\sum_{o \in \mathcal{S}^t} \min_{\mathcal{C} \in \Phi} \mathcal{E}(o, \mathcal{C})}{|\mathcal{S}^t|}$$

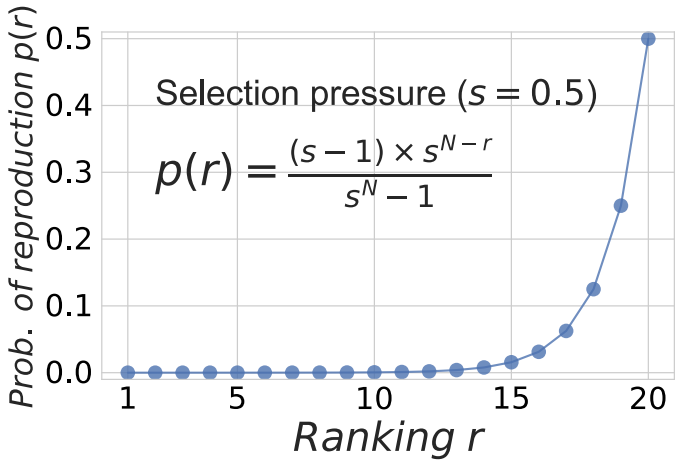


Selection

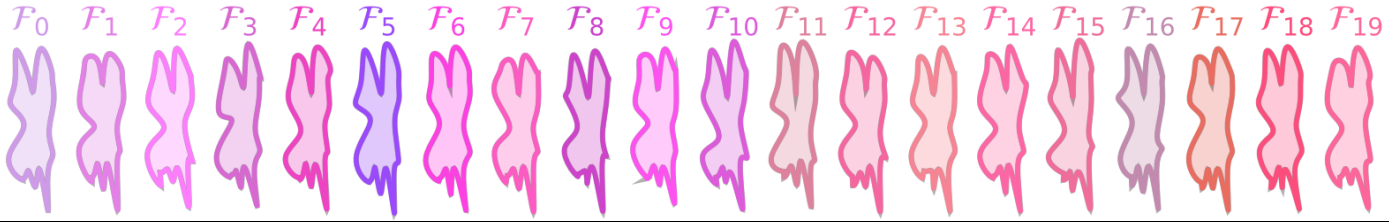
- Generation **t** (order Individuals by fitness: Highest rank → best):



- **Exponential ranking** selection:



- Mutate children → Generation **t+1**:

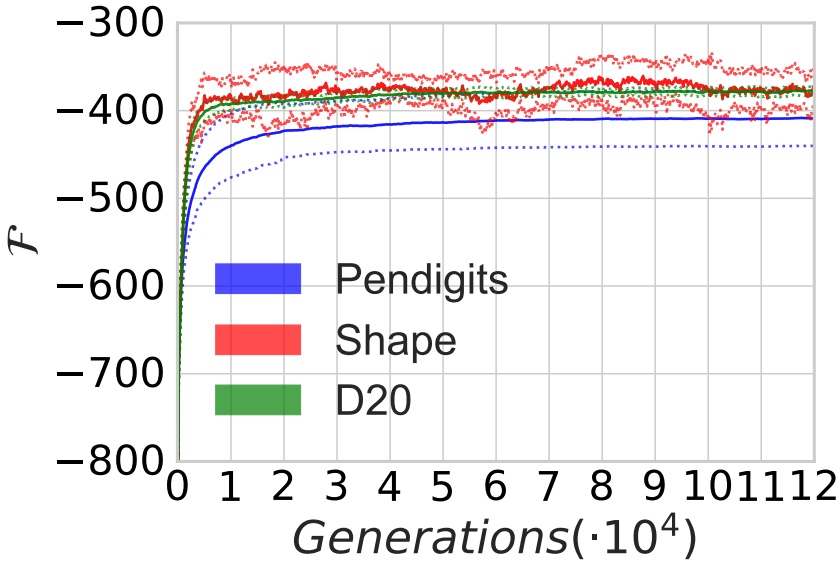


Evaluating the evolutionary algorithm

- **Compare** Chameleoclust to **state-of-the-art algorithms**.
- Reference **evaluation framework** [Müller et al. VLDB 2009]:
  - **7 real** benchmark datasets:
    - *shape*, *pendigitis*, *liver*, *glass*, *breast*, *diabetes*, *vowel*
  - **16 synthetic** benchmark datasets with different:
    - **Nb. of dimensions:** *D05*, *D10*, *D15*, *D20*, *D25*, *D50*, *D75*.
    - **Nb. of objects:** *S1500*, *S2500*, *S3500*, *S4500*, *S5500*.
    - **Percentage of noise objects:** *N10*, *N30*, *N50*, *N70*.

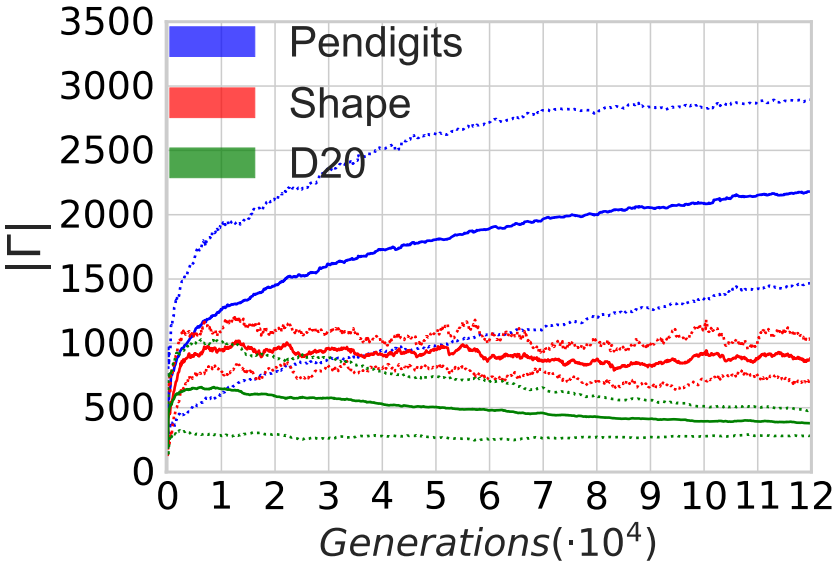
Evolution of the organisms

Fitness

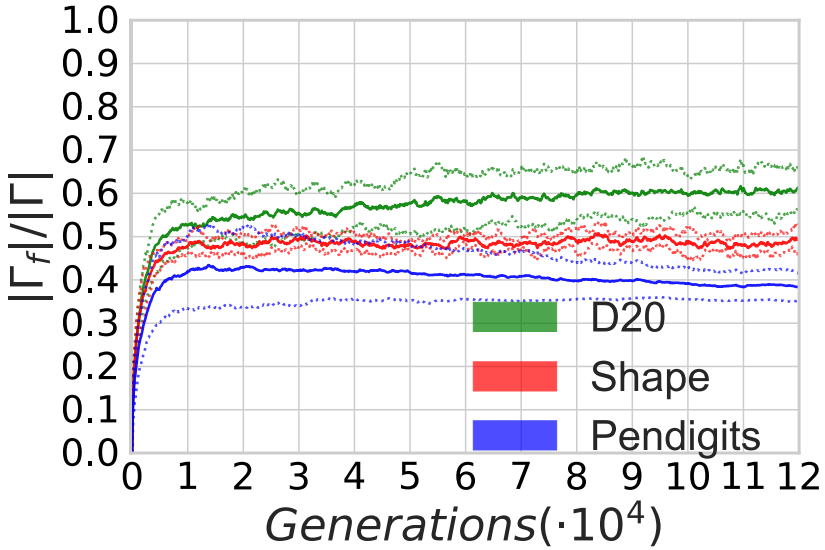


- 10 runs over each dataset.
- Mean $\pm$ Standard deviation.

Genome size



Functional ratio

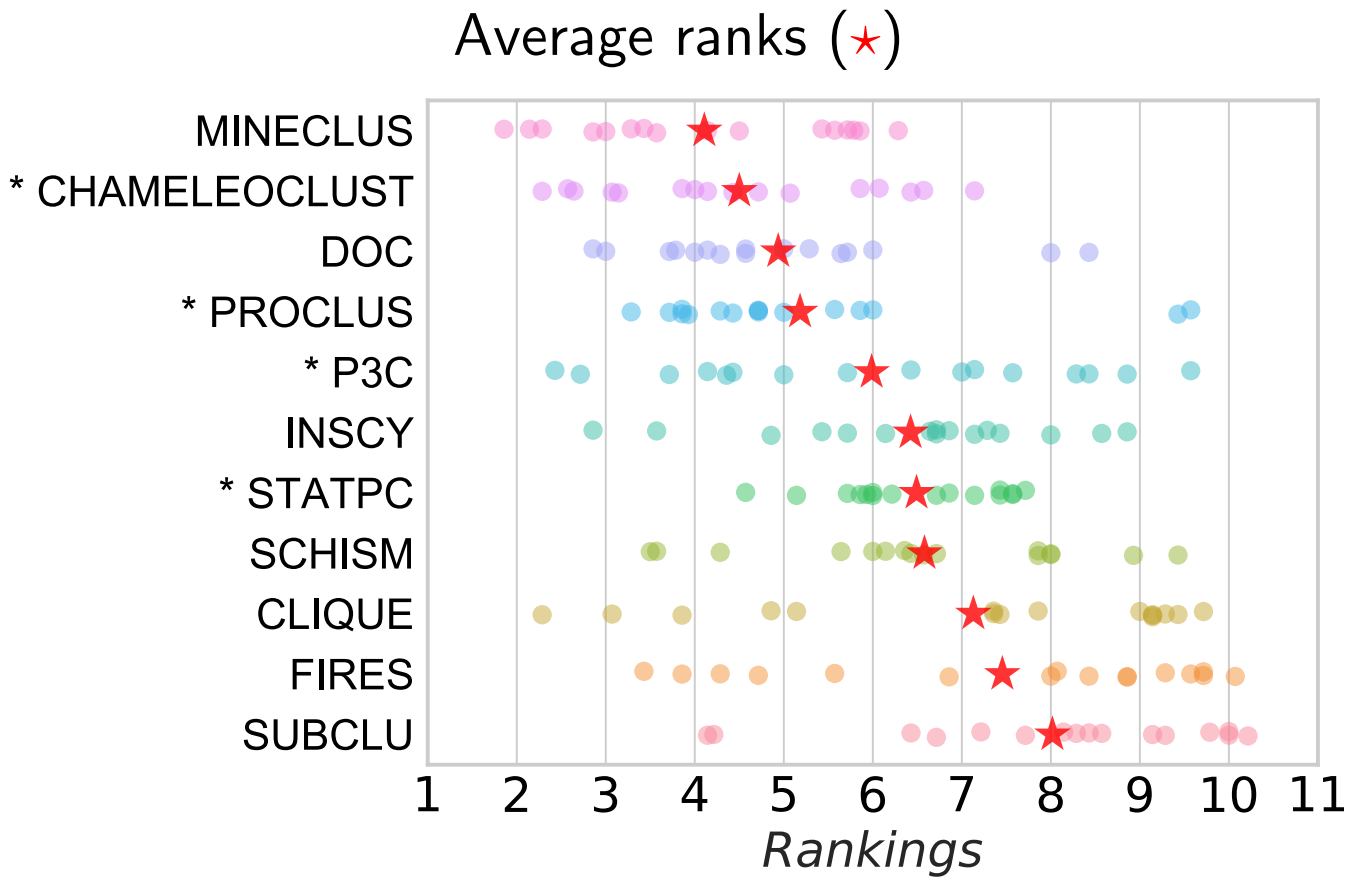


Results : Real data (e.g., Shape dataset)

		CLIQUE	*DOC*	*MINECLUS*	*SCHISM*	*SUBCLU*	*FIRES*	*INSCY*	*PROCLUS**	*P3C**	*STATPC**	*Chameleoclust**
Accuracy	*max*	0.76	0.79	0.79	0.74	0.70	0.51	0.76	0.72	0.61	0.74	0.8
	min	0.76	0.54	0.60	0.49	0.64	0.44	0.48	0.71	0.61	0.74	0.71
SubspaceCE	*max*	0.01	0.56	0.58	0.10	0.00	0.20	0.18	0.25	0.14	0.45	0.54
	min	0.01	0.38	0.46	0.00	0.00	0.13	0.16	0.18	0.14	0.45	0.49
NumClusters	*max*	486	53	64	8835	3468	10	185	34	9	9	14
	min	486	29	32	90	3337	5	48	34	9	9	10
AvgDim	*max*	3.3	13.8	17.0	6.0	4.5	7.6	9.8	13.0	4.1	17	12.4
	min	3.3	12.8	17.0	3.9	4.1	5.3	9.5	7.0	4.1	17	10.8
RunTime	*max*	235	2E+06	46703	712964	4063	63	22578	593	140	250	462
	min	235	86500	3266	9031	1891	47	11531	469	140	171	252

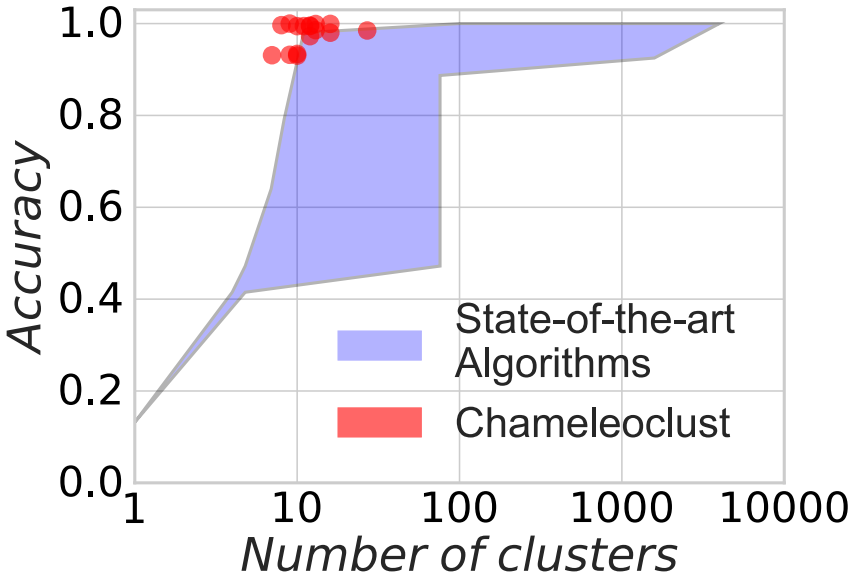
- Algorithms belonging to the **Axis-parallel cluster-oriented family** (*).
- **Competitive** performances with respect to other algorithms.
- Good **compromise** between the **different evaluation measures**.

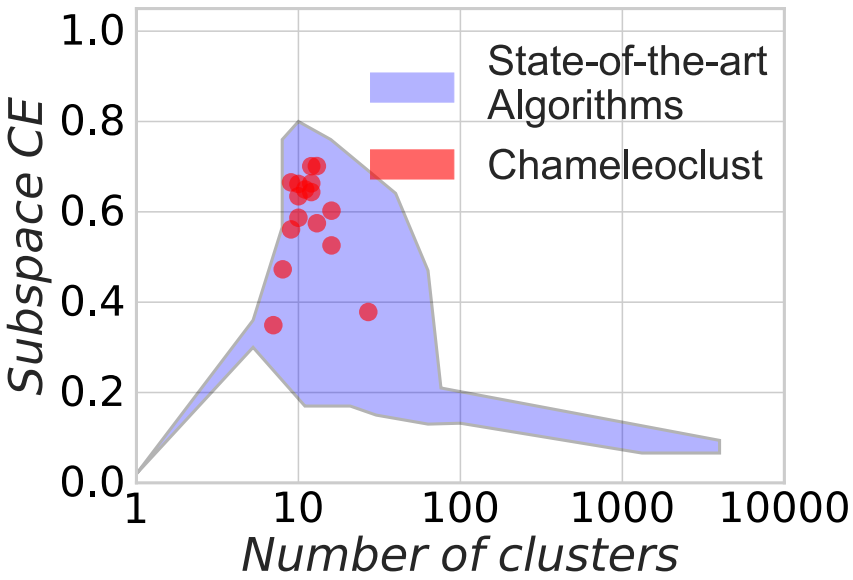
Results: Real datasets



- **Ranks** of the algorithms regarding different **quality measures** (*Accuracy*, *Subspace CE*, F_1, $\overline{Entropy}$, $\overline{RNIA}$, *Coverage*, *NumClusters*, *AvgDim*, *RunTime*) for the **7 real datasets**.
- **Competitive** performances and good **compromise**.

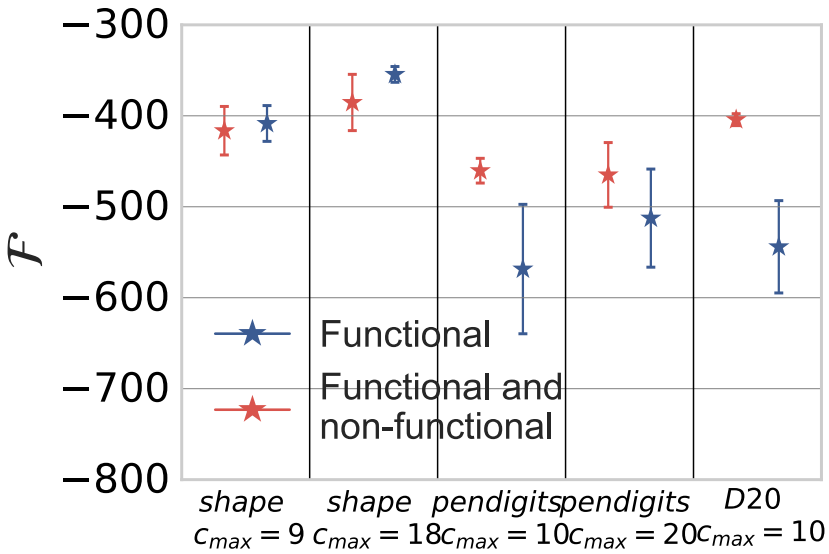
Results: Synthetic datasets

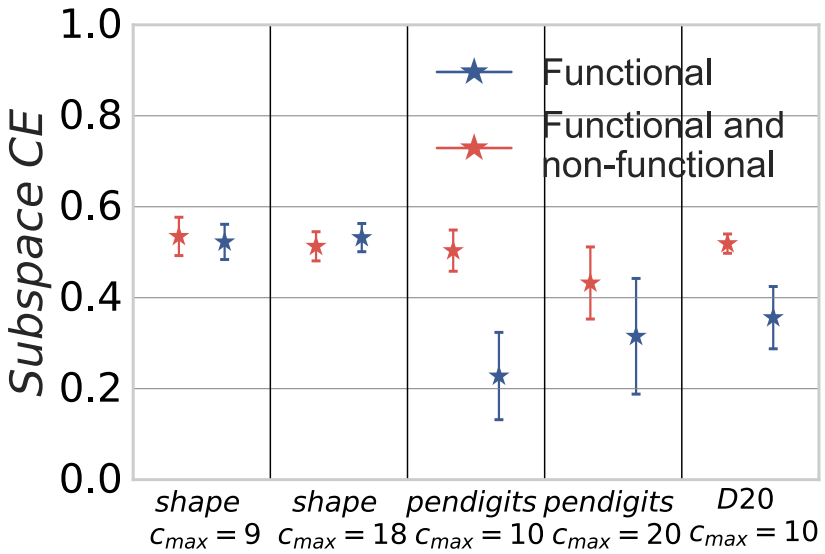




- 16 synthetic datasets.
- **True number of clusters = 10**.
- **Best quality = 1**.
- Good **compromize** between the **number of clusters found** and the **quality**.

Results: Impact of non-functional tuples





- **Positive** impact of **non-functional elements**.

Outline

SubMorphoStream

- **Cluster-oriented** subspace clustering of **data streams**.
- More **conceptual representation** based on **tandem arrays**.
- Bio-inspired operators adapted to **changing environments**.

SubMorphoStream

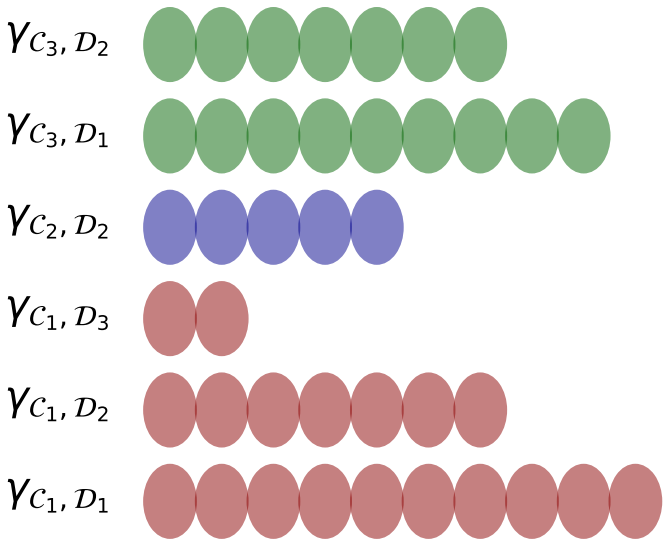
- **Cluster-oriented** subspace clustering of **data streams**.
- More **conceptual representation** based on **tandem arrays**.
- Bio-inspired operators adapted to **changing environments**.
  - **Amplification** and **Deamplification**.

SubMorphoStream

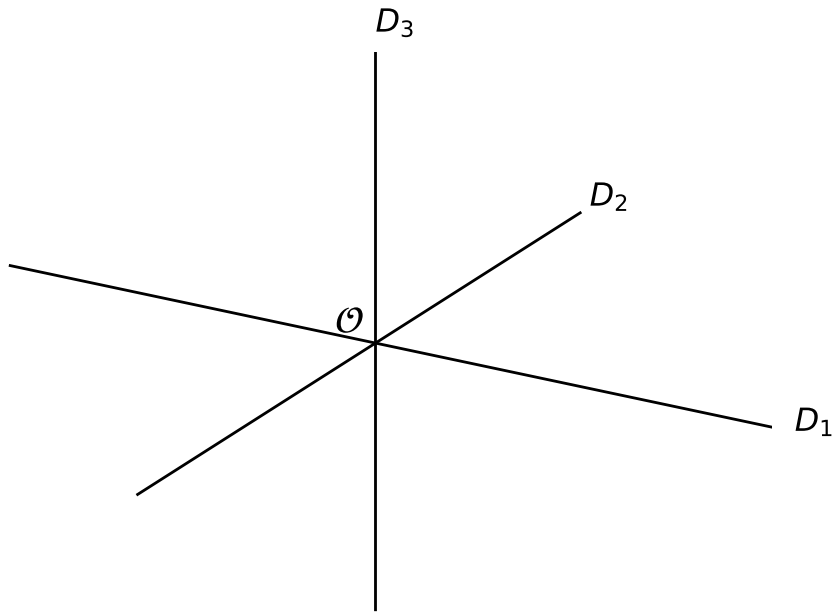
- **Cluster-oriented** subspace clustering of **data streams**.
- More **conceptual representation** based on **tandem arrays**.
- Bio-inspired operators adapted to **changing environments**.
  - **Amplification** and **Deamplification**.
  - **Exogenous** genetic uptake

Genotype to Phenotype mapping

Genotype: Γ



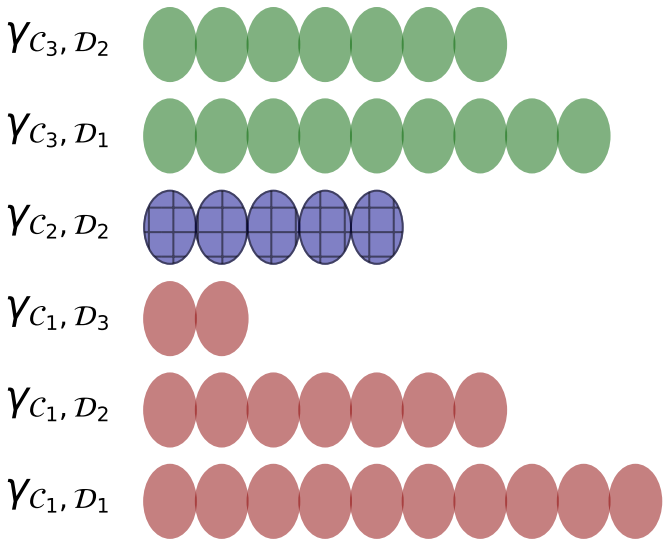
Phenotype: Φ



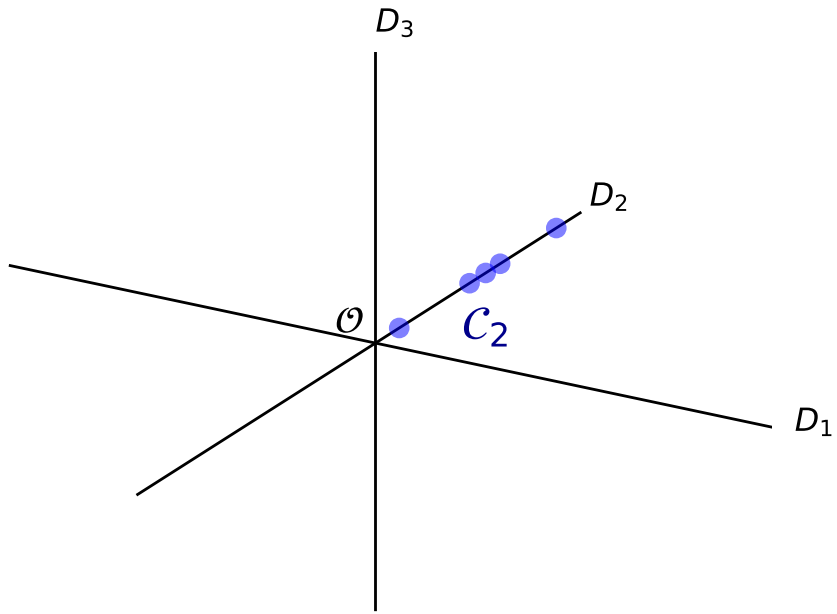
- Γ: bag of **tandem arrays** $\gamma_{\mathcal{C},\mathcal{D}}$.

Genotype to Phenotype mapping

Genotype: Γ



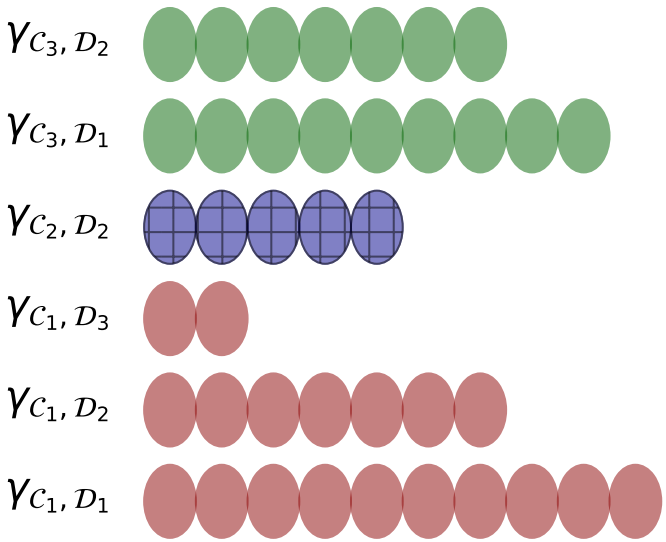
Phenotype: Φ



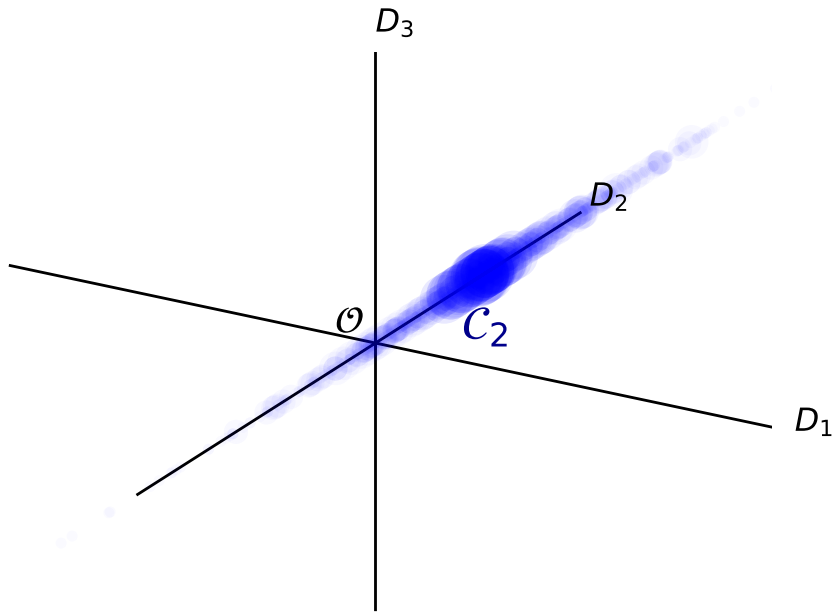
- **Genes** in $\gamma_{\mathcal{C},\mathcal{D}}$ contribute to **core-point** $\mathcal{C}$ along **dimension** $\mathcal{D}$

Genotype to Phenotype mapping

Genotype: Γ

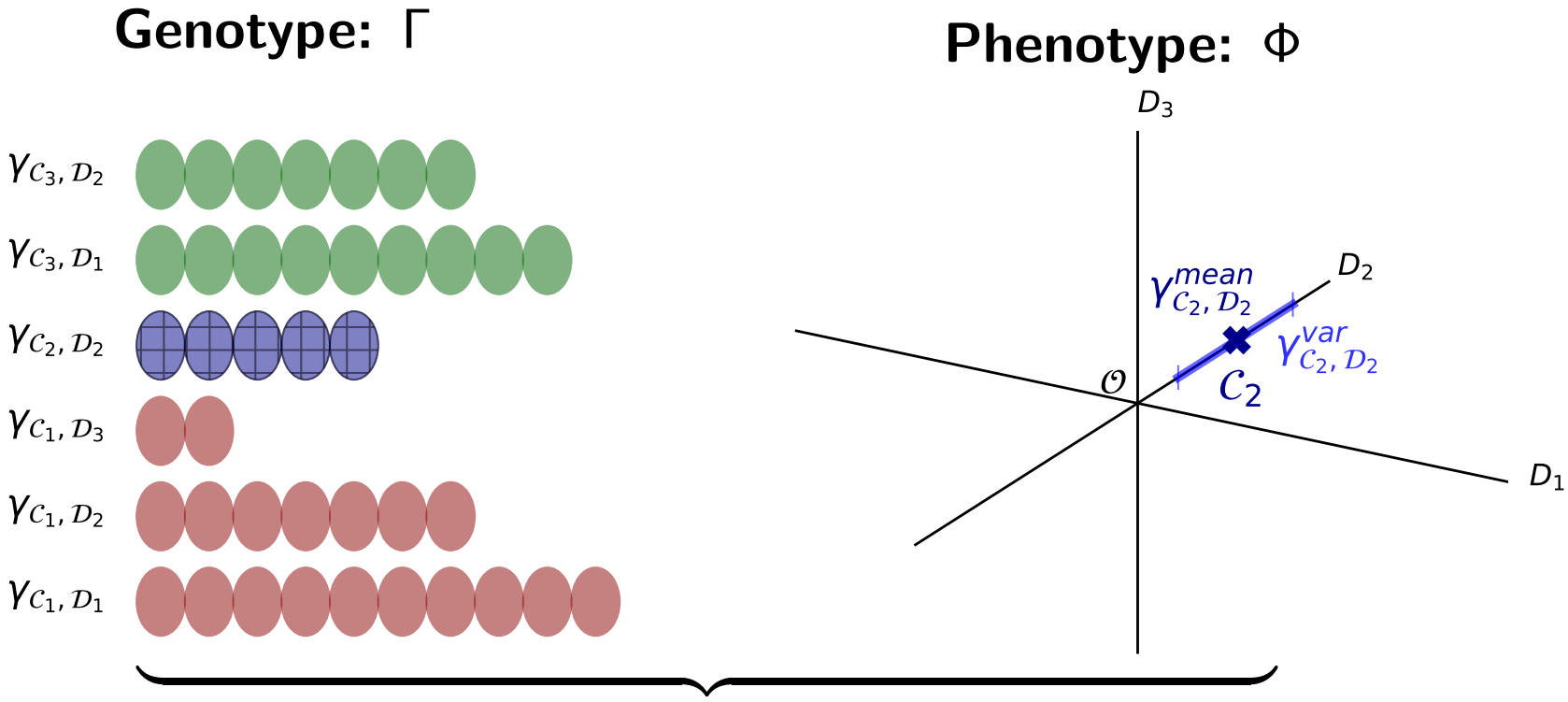


Phenotype: Φ



- Genes are **not** encoded explicitly.
- Assumption: **contributions** follow a **gaussian distribution**.

Genotype to Phenotype mapping

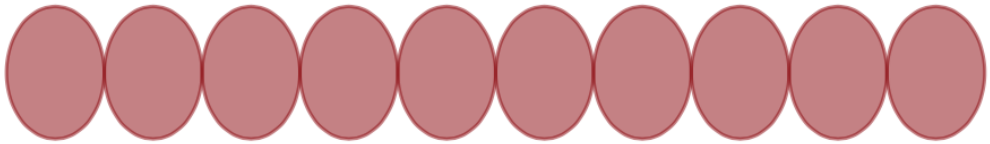


$$\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$$

Joint representation of the **phenotype** and the **genotype**.

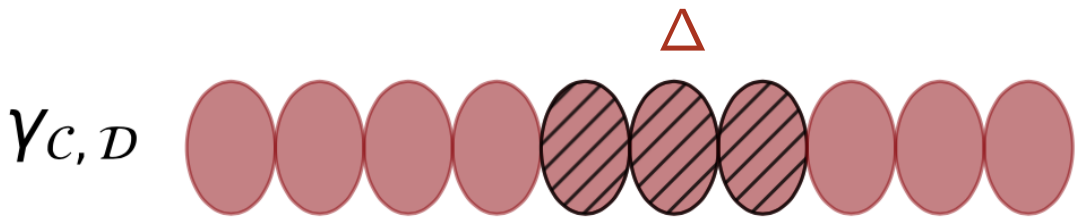
- $\gamma^{size}_{\mathcal{C},\mathcal{D}}$: **Size** of the tandem array (number of genes).
- $\gamma^{mean}_{\mathcal{C},\mathcal{D}}$: **Mean** contribution: $\gamma^{mean}_{\mathcal{C}_2,\mathcal{D}_2}$ (**core-point locations**).
- $\gamma^{var}_{\mathcal{C},\mathcal{D}}$: **Variance** of the contributions.

Mutational operators
Amplification/Deamplifications

$\gamma_{\mathcal{C},\mathcal{D}}$ 

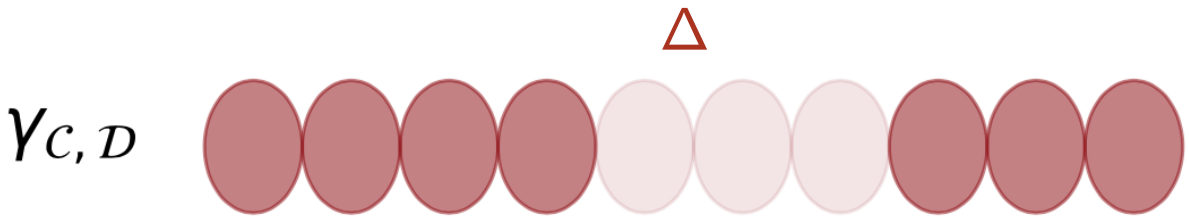
- $\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$

Mutational operators
Amplification/Deamplifications



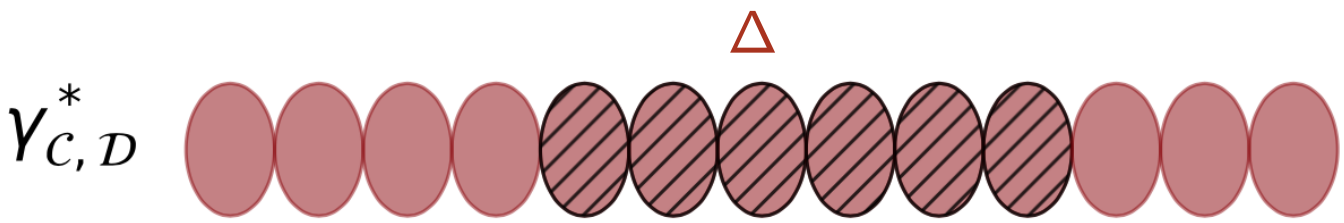
- $\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
- $\Delta = \langle {}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$

Mutational operators
Amplification/Deamplifications



- $\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
- $\Delta = \langle {}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
  - Uniform number of Deleted
    ${}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}} \sim \mathcal{U}(\underbrace{\{-\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, -1\}}_{Deamplification})$

Mutational operators
Amplification/Deamplifications



- $\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
- $\Delta = \langle {}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
  - **Uniform** number of deleted or duplicated elements
    ${}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}} \sim \mathcal{U}(\underbrace{\{-\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, -1\}}_{\textit{Deamplification}} \cup \underbrace{\{\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, 1\}}_{\textit{Amplification}})$

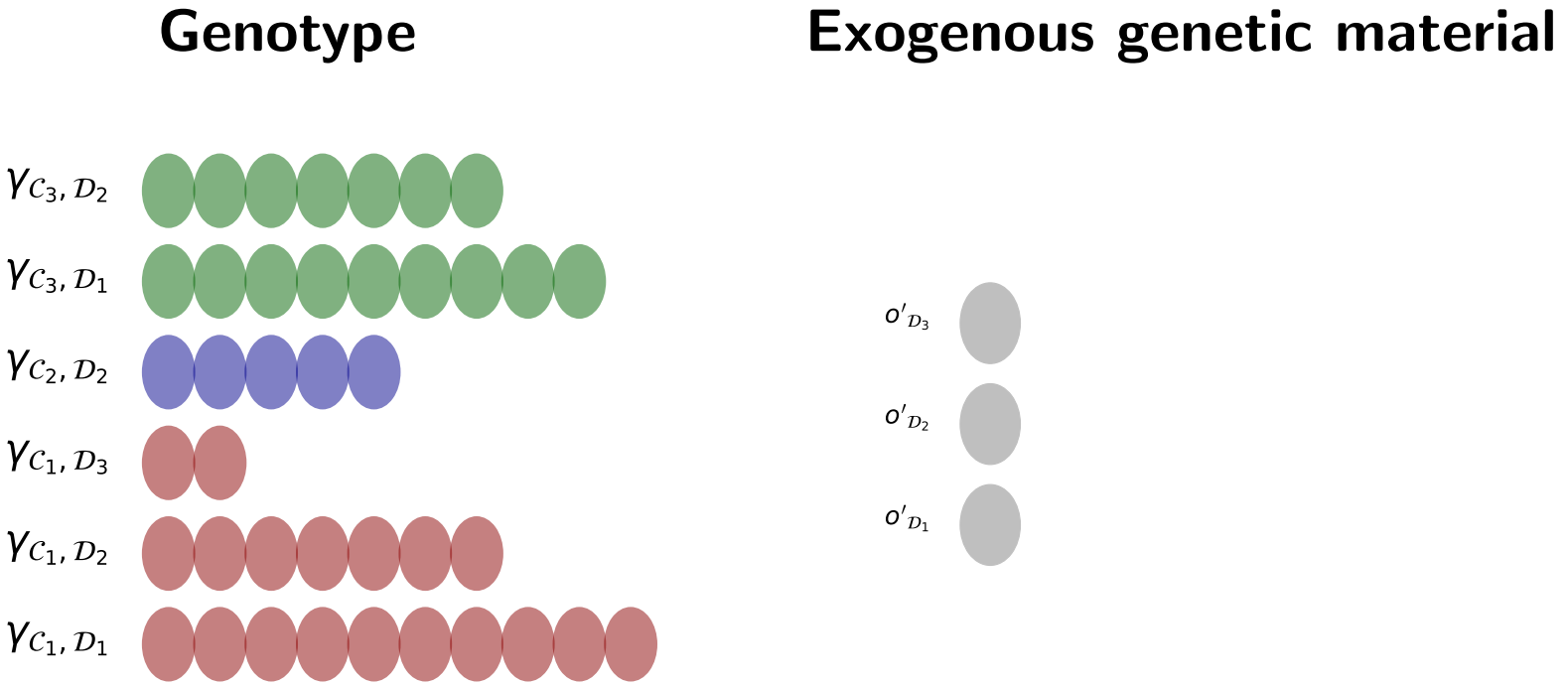
Mutational operators
Amplification/Deamplifications

- $\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
- $\Delta = \langle {}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
  - **Uniform** number of deleted or duplicated elements
    ${}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}} \sim \mathcal{U}(\underbrace{\{-\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, -1\}}_{Deamplification} \cup \underbrace{\{\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, 1\}}_{Amplification})$
  - **Mean of a sample** of normally distributed values.
    ${}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}} \sim \mathcal{N}(\gamma^{mean}_{\mathcal{C},\mathcal{D}}, \frac{\gamma^{var}_{\mathcal{C},\mathcal{D}}}{|{}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}|})$

- $\gamma_{\mathcal{C},\mathcal{D}} = \langle \gamma^{size}_{\mathcal{C},\mathcal{D}}, \gamma^{mean}_{\mathcal{C},\mathcal{D}}, \gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
- $\Delta = \langle {}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}}, {}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} \rangle$
  - **Uniform** number of deleted or duplicated elements
    ${}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}} \sim \mathcal{U}(\underbrace{\{-\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, -1\}}_{Deamplification} \cup \underbrace{\{\gamma^{size}_{\mathcal{C},\mathcal{D}}, \ldots, 1\}}_{Amplification})$
  - **Mean of a sample** of normally distributed values.
    ${}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}} \sim \mathcal{N}(\gamma^{mean}_{\mathcal{C},\mathcal{D}}, \frac{\gamma^{var}_{\mathcal{C},\mathcal{D}}}{|{}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}|})$
  - **Variance of a sample** of normally distributed values.
    ${}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} \times \frac{(|{}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}|-1)}{\gamma^{var}_{\mathcal{C},\mathcal{D}}} \sim \chi^2_{|{}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}|-1}$

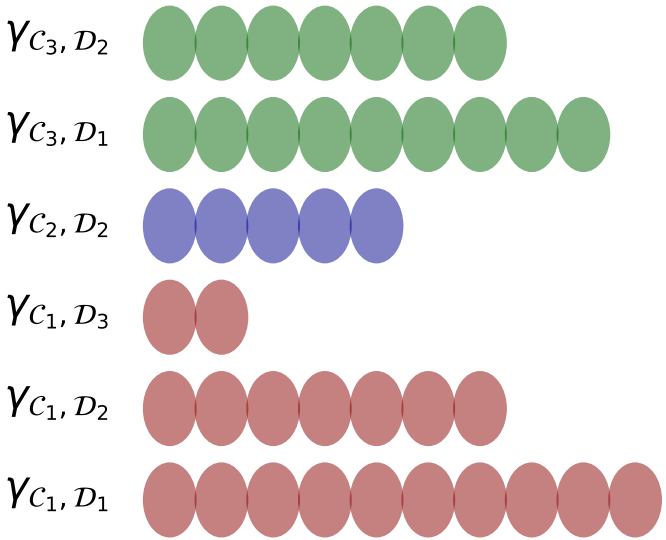
- Update tandem array: $\gamma_{\mathcal{C},\mathcal{D}} \leftarrow \gamma^*_{\mathcal{C},\mathcal{D}}$
  - **Update** the tandem array **size**.
    $\gamma^{size\ *}_{\mathcal{C},\mathcal{D}} = \gamma^{size}_{\mathcal{C},\mathcal{D}} + {}^{\Delta}\gamma^{size}_{\mathcal{C},\mathcal{D}}$
  - **Incremental update** of the **mean** of the contributions.
    $\gamma^{mean*}_{\mathcal{C},\mathcal{D}} = \frac{1}{\gamma^{size\ *}_{\mathcal{C},\mathcal{D}}} \times (\gamma^{mean}_{\mathcal{C},\mathcal{D}} \times \gamma^{size}_{\mathcal{C},\mathcal{D}} + {}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}} \times^{\Delta} \gamma^{size}_{\mathcal{C},\mathcal{D}})$
  - **Incremental update** of the **variance** of the contributions.
    $\gamma^{var\ *}_{\mathcal{C},\mathcal{D}} = \frac{\gamma^{size}_{\mathcal{C},\mathcal{D}}}{\gamma^{size\ *}_{\mathcal{C},\mathcal{D}}} \times (\gamma^{var}_{\mathcal{C},\mathcal{D}} + (\gamma^{mean}_{\mathcal{C},\mathcal{D}} - \gamma^{mean*}_{k,d})^2) +$
    $\frac{{}^{\Delta}\gamma^{size}_{k,d}}{\gamma^{size\ *}_{k,d}} \times ({}^{\Delta}\gamma^{var}_{\mathcal{C},\mathcal{D}} + ({}^{\Delta}\gamma^{mean}_{\mathcal{C},\mathcal{D}} - \gamma^{mean*}_{\mathcal{C},\mathcal{D}})^2)$

Mutational operators
Exogenous gene uptake

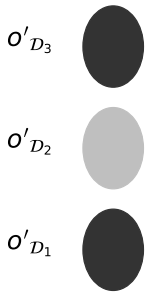


Mutational operators
Exogenous gene uptake

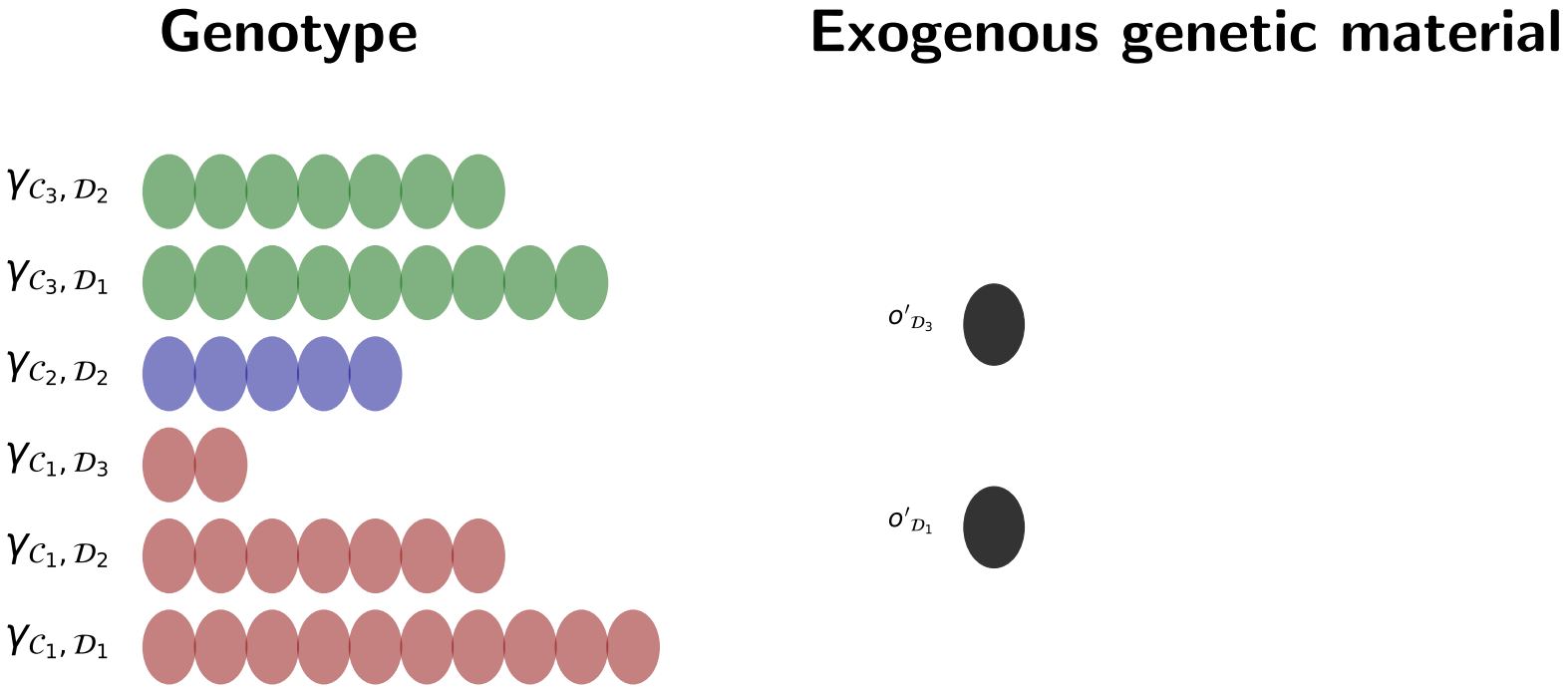
Genotype



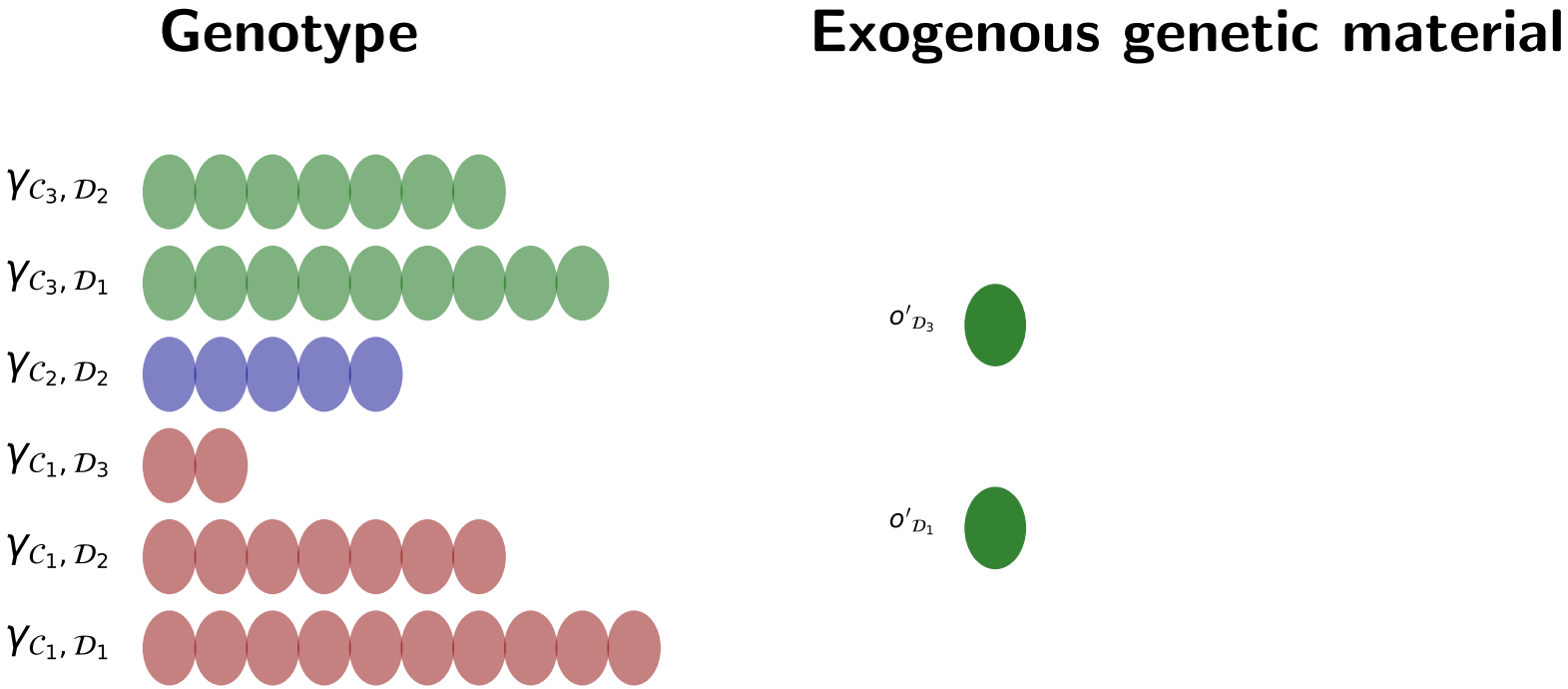
Exogenous genetic material



Mutational operators
Exogenous gene uptake

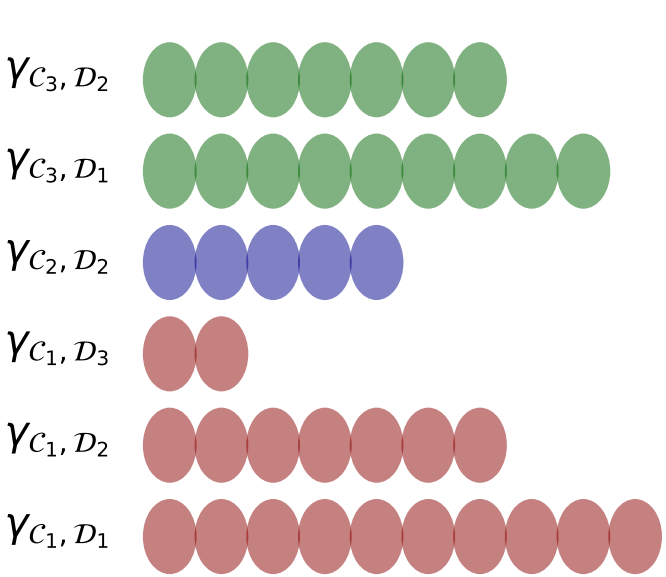


Mutational operators
Exogenous gene uptake

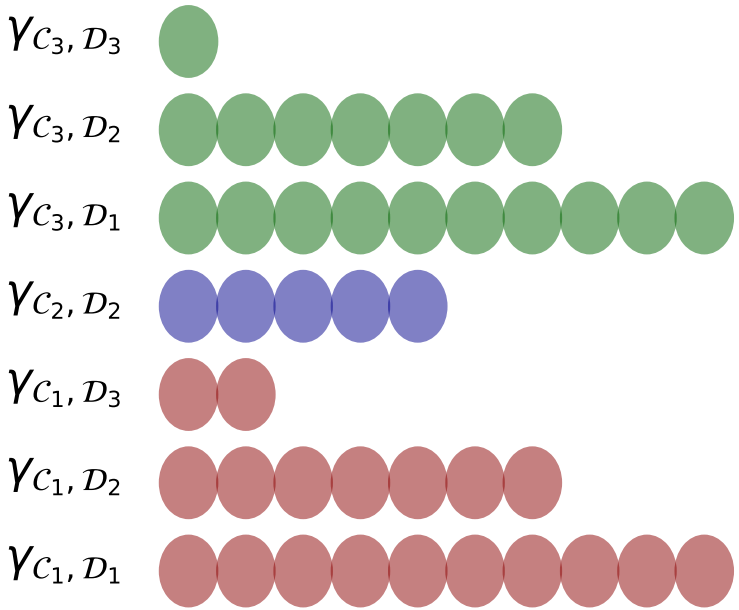


Mutational operators
Exogenous gene uptake

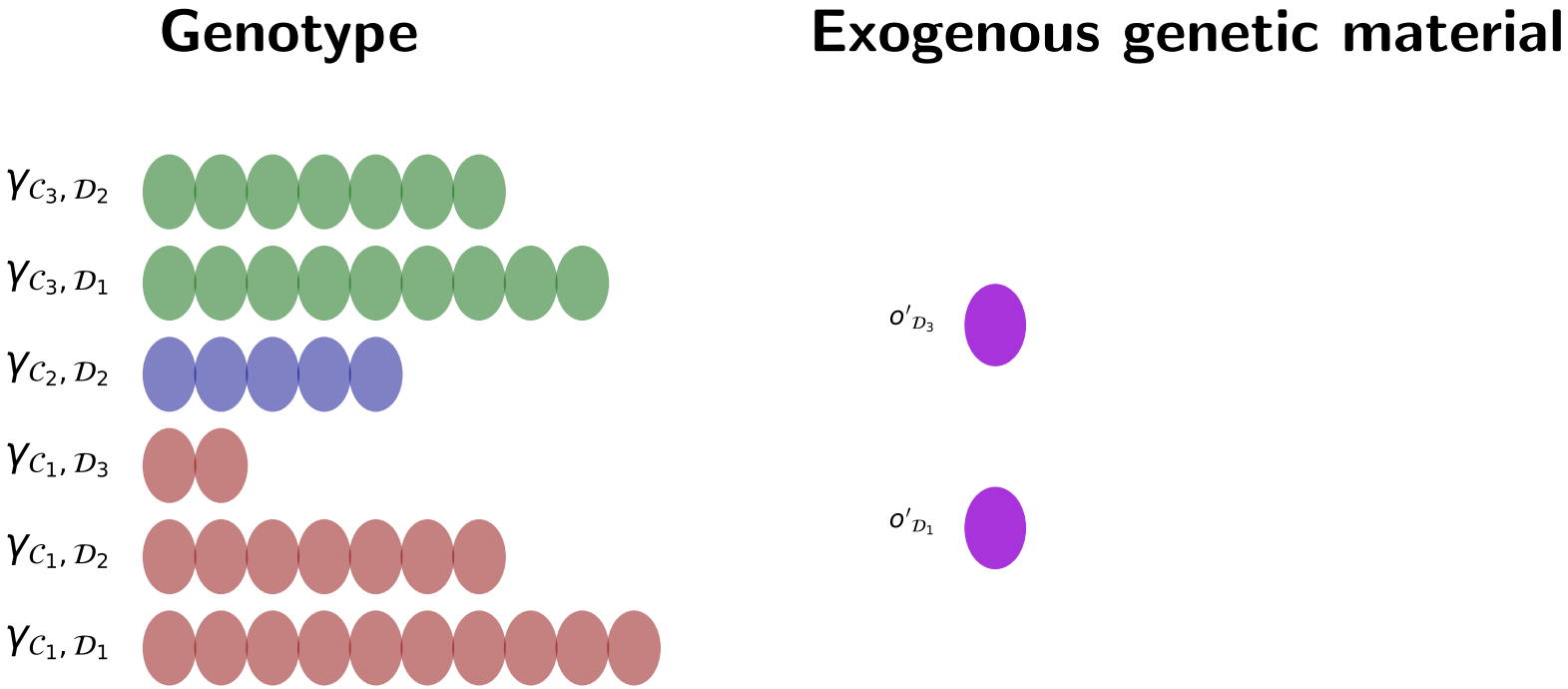
Genotype



Genotype after mutation

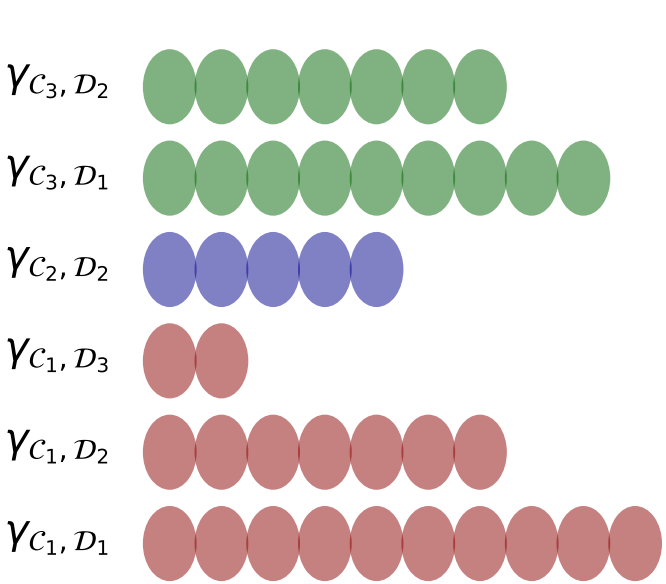


Mutational operators
Exogenous gene uptake

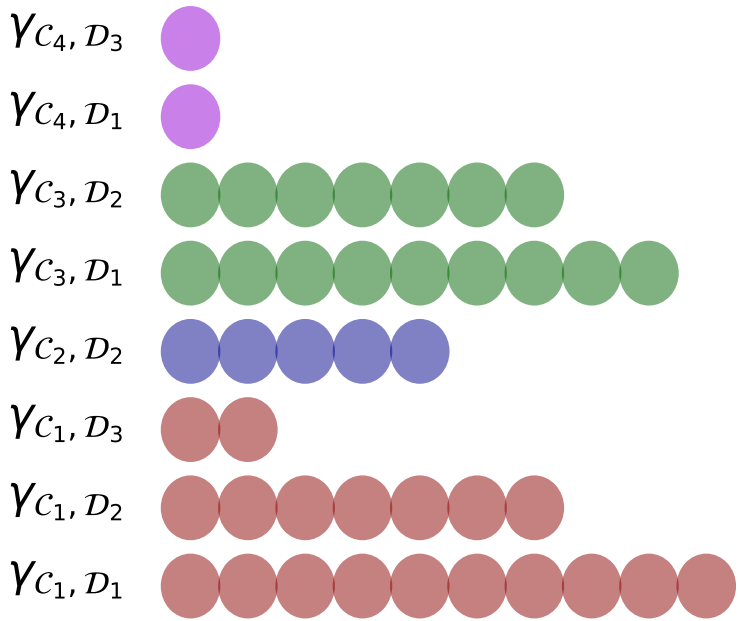


Mutational operators
Exogenous gene uptake

Genotype



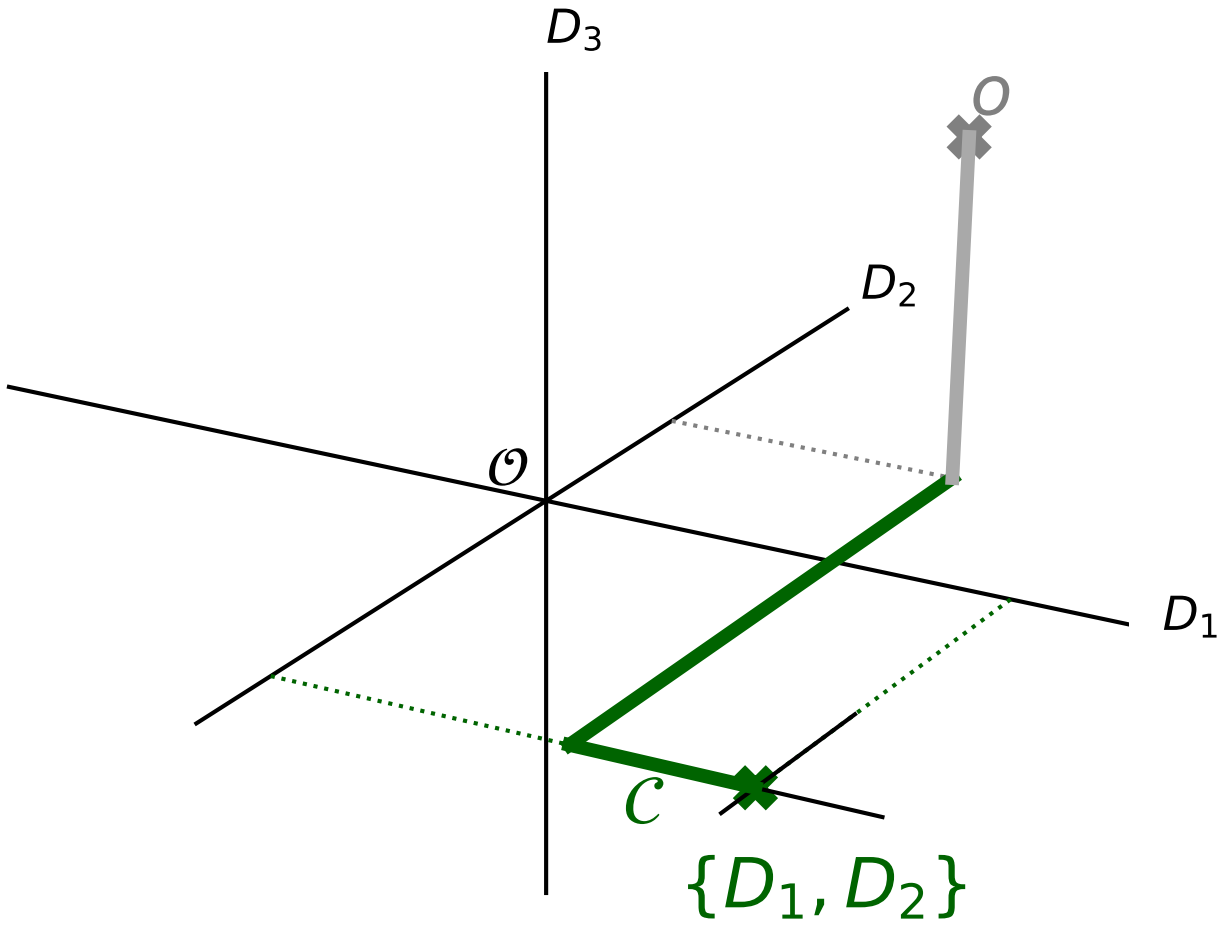
Genotype after mutation



- Update tandem array: $\gamma_{\mathcal{C},\mathcal{D}} \leftarrow \gamma^*_{\mathcal{C},\mathcal{D}}$
  - **Update** the tandem array **size**.
    $\gamma^{size}_{\mathcal{C},\mathcal{D}}{}^* = \gamma^{size}_{\mathcal{C},\mathcal{D}} + 1$
  - **Incremental update** of the **mean** of the contributions.
    $\gamma^{mean}_{\mathcal{C},\mathcal{D}}{}^* = \frac{\gamma^{mean}_{\mathcal{C},\mathcal{D}} \times \gamma^{size}_{\mathcal{C},\mathcal{D}} + o'_{\mathcal{D}}}{\gamma^{size}_{\mathcal{C},\mathcal{D}} + 1}$
  - **Incremental update** of the **variance** of the contributions.
    $\gamma^{var}_{\mathcal{C},\mathcal{D}}{}^* = \frac{\gamma^{size}_{\mathcal{C},\mathcal{D}} \times \gamma^{var}_{\mathcal{C},\mathcal{D}}}{\gamma^{size}_{\mathcal{C},\mathcal{D}} + 1} + \frac{\gamma^{size}_{\mathcal{C},\mathcal{D}} \times (\gamma^{mean}_{\mathcal{C},\mathcal{D}} - o'_{\mathcal{D}})^2}{(\gamma^{size}_{\mathcal{C},\mathcal{D}} + 1)^2}$

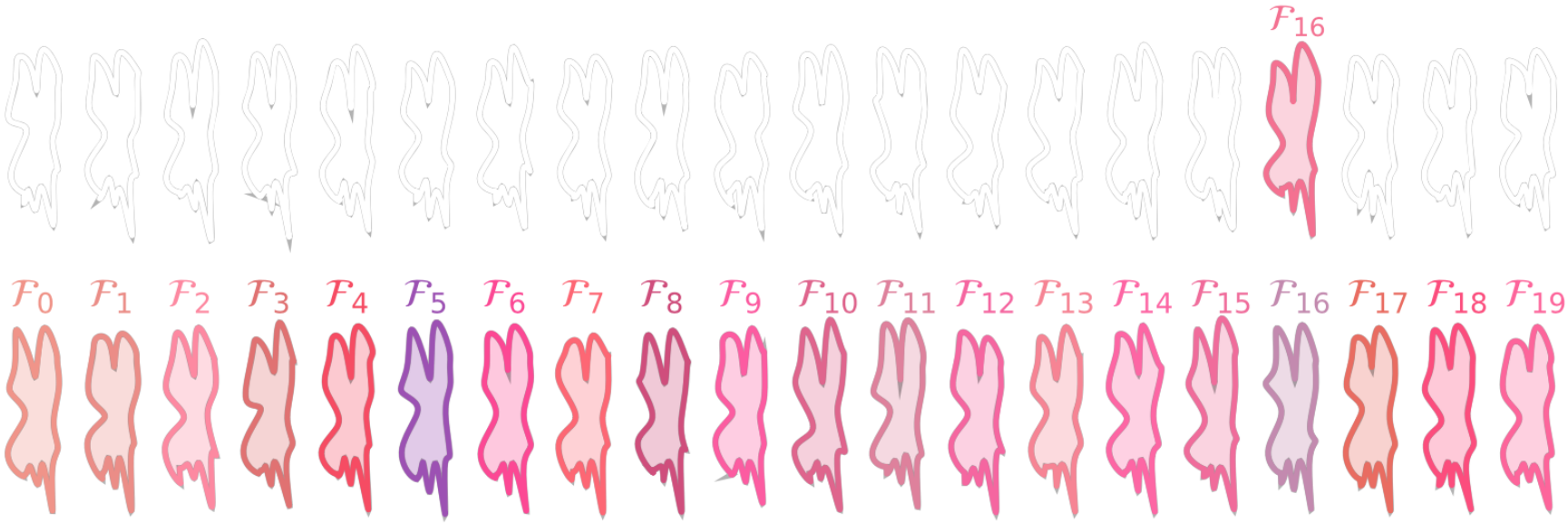
Fitness computation

- Fitness **analogous** to the one of **Chameleoclust**.
- **Assign** each object in window $\mathcal{S}^t$ to its **closest** core-point $\mathcal{C} \in \Phi$ (**Manhattan distance**).
- Sum of distances between **data objects** and **core-points**.



Selection

- $(1+\lambda)$-ES selection scheme.
  - 1 parental organism.
  - λ children.



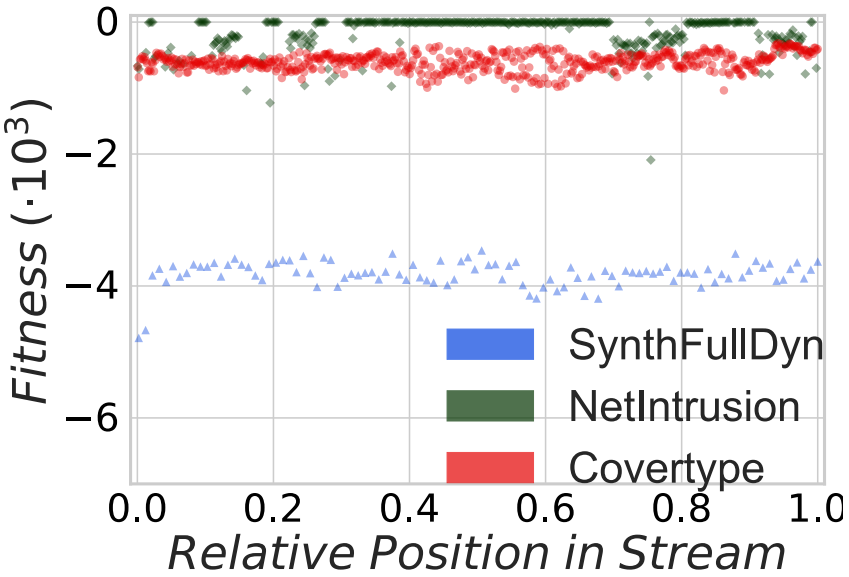
Evaluating the evolutionary algorithm

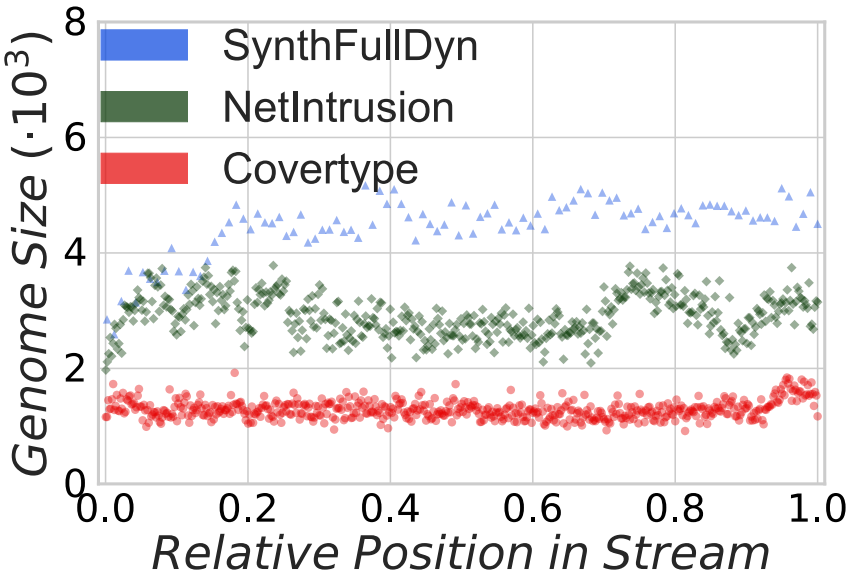
- **Compare** SubMorphoStream to the state-of-the-art algorithm HPStream.

- **2 real** benchmark datasets
  (*Network intrusion* and *Forest cover*).

- **6 synthetic** benchmark datasets.
  - *SynthBaseDyn* : Dimensions importance (+).
  - *SynthClusterSizeDyn* : Cluster sizes.
  - *SynthFeatureDyn* : Dimensions importance (++).
  - *SynthClusterNbDyn* : Clusters appearing/disappearing.
  - *SynthDriftDyn* : Clusters drift.
  - *SynthFullDyn* : All changes at once.

Evolution of the organisms

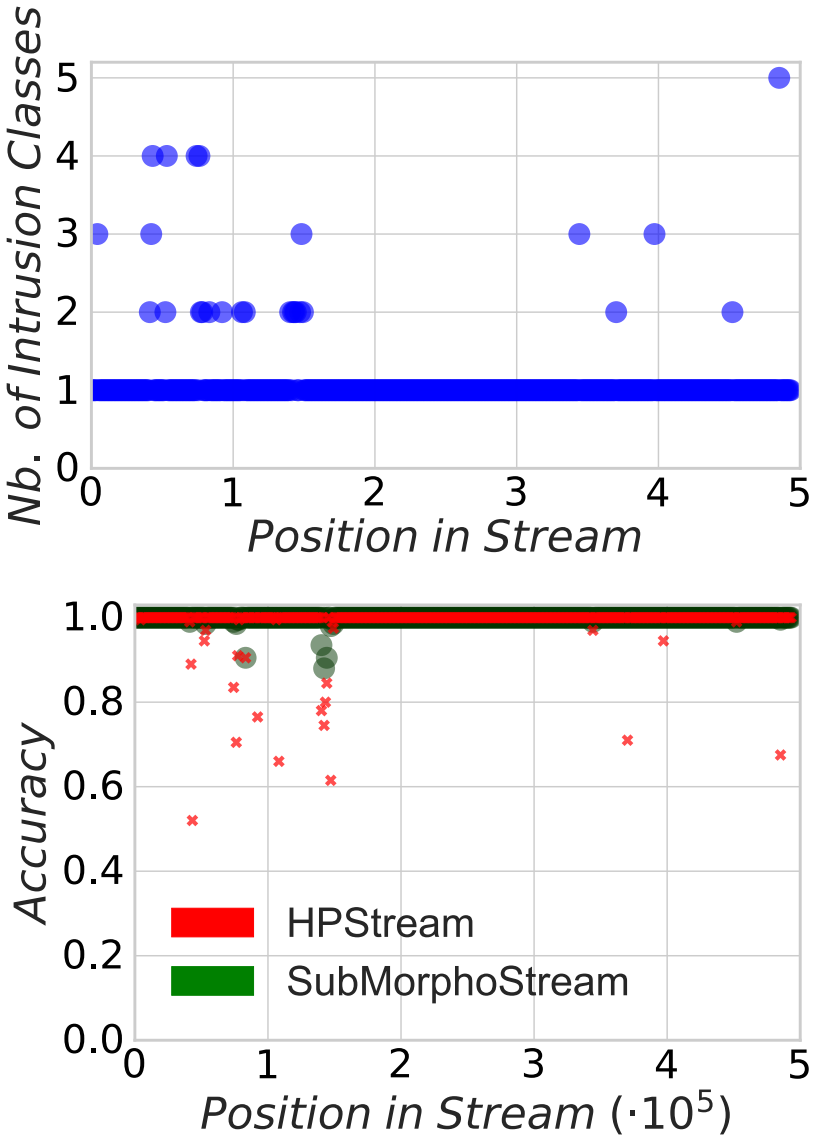




- **Dynamic evolution** of the **fitness** and the **genome structure**.
- **Adaptation** to each **data stream**.

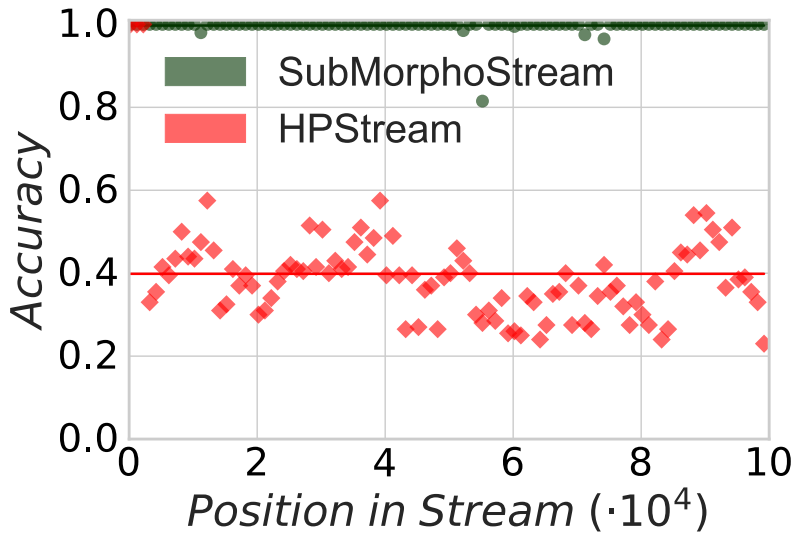
Results: Real datasets

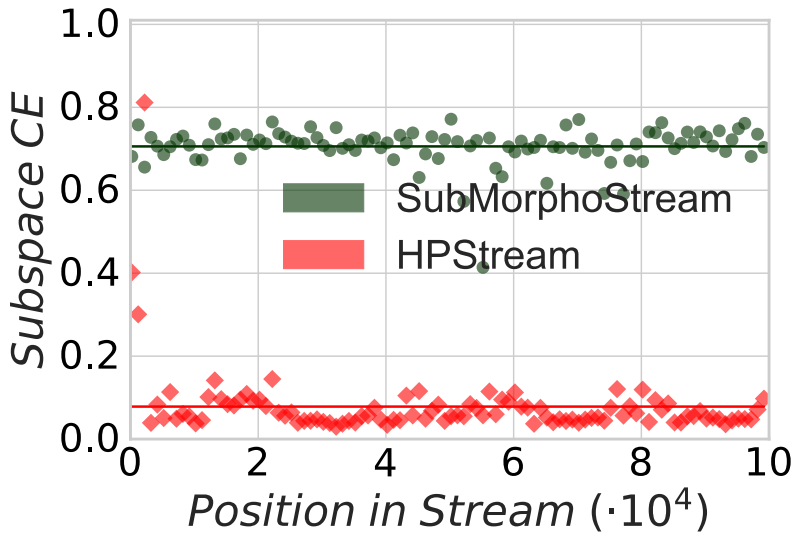
Network Intrusion dataset

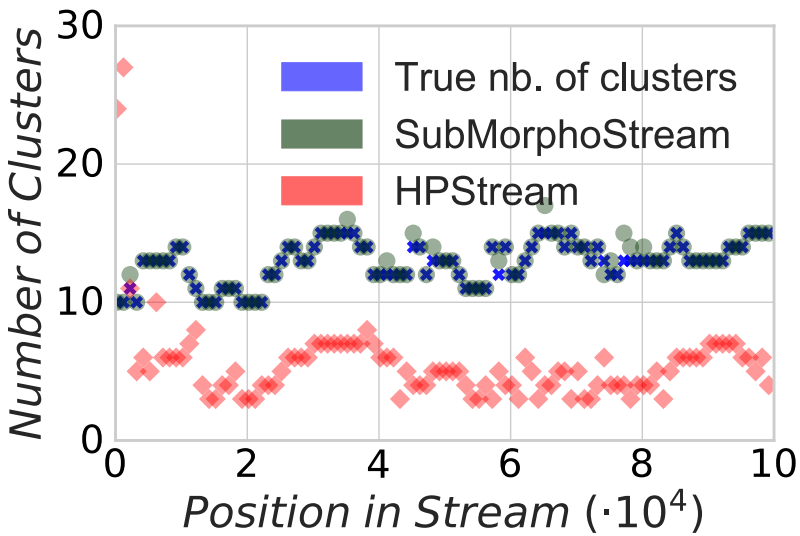


- SubMorphoStream is more **robust** to **changes** in the stream.

Results: Synthetic dataset (*SynthFullDyn*)







- Very dynamic data stream:
  - Clusters appearance and disappearance.
  - Clusters drifting.
  - Cluster sizes.
  - Dimensions importance.
- SubMorphoStream exhibits better cluster identification.

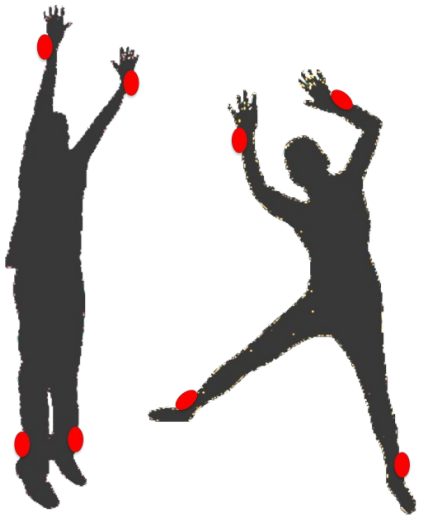
Results: Synthetic datasets

	Accuracy		Subspace CE	
	HPStream	SubMorphoStream	HPStream	SubMorphoStream
SynthBaseDyn	0.999 ± 0.004	1.0 ± 0.0	0.3 ± 0.043	0.617 ± 0.099
SynthClusterSizeDyn	1.0 ± 0.0	1.0 ± 0.0	0.318 ± 0.045	0.605 ± 0.101
SynthFeatureDyn	1.0 ± 0.001	1.0 ± 0.0	0.426 ± 0.076	0.597 ± 0.106
SynthClusterNbDyn	0.843 ± 0.099	0.992 ± 0.044	0.488 ± 0.142	0.685 ± 0.104
SynthDriftDyn	0.452 ± 0.122	1.0 ± 0.0	0.105 ± 0.084	0.716 ± 0.035
SynthFullDyn	0.398 ± 0.133	0.997 ± 0.019	0.078 ± 0.088	0.706 ± 0.047

- Not significantly different results.
- Significantly higher results.
- Significantly lower results.

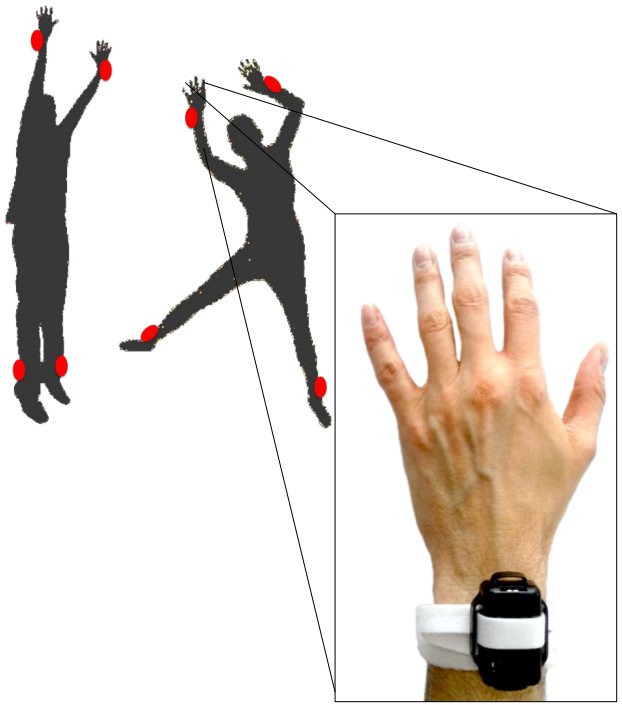
Outline

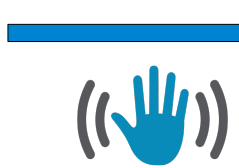
EvoMove System.

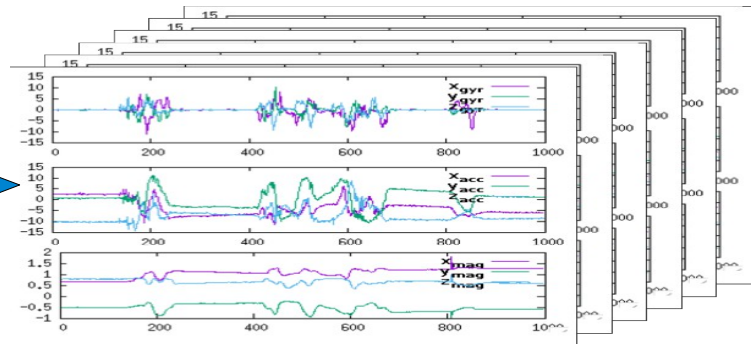


EvoMove System.

Dancer(s)

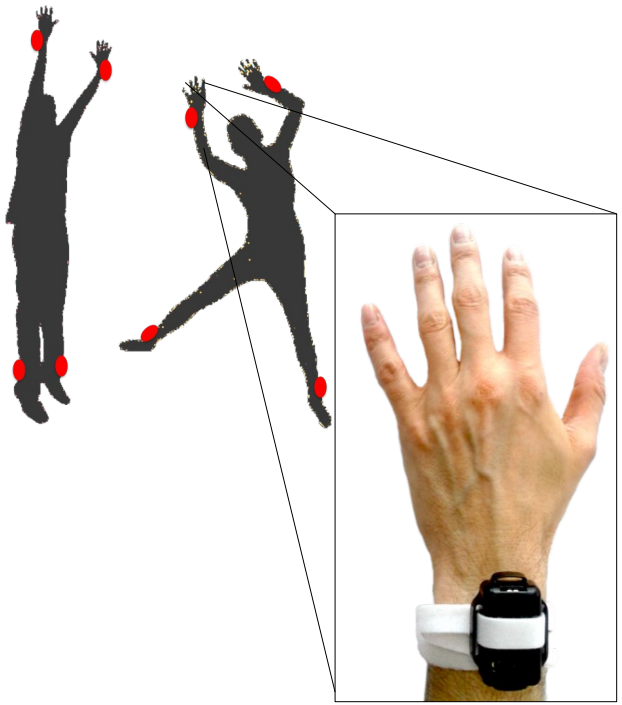


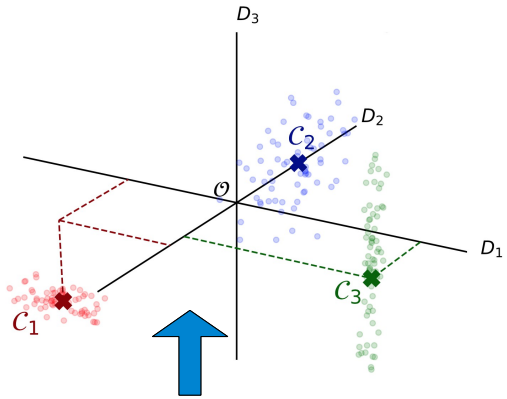




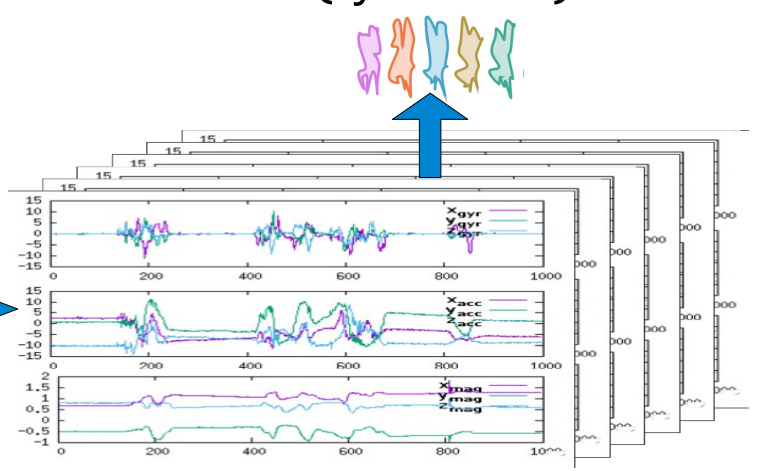
EvoMove System.

Dancer(s)

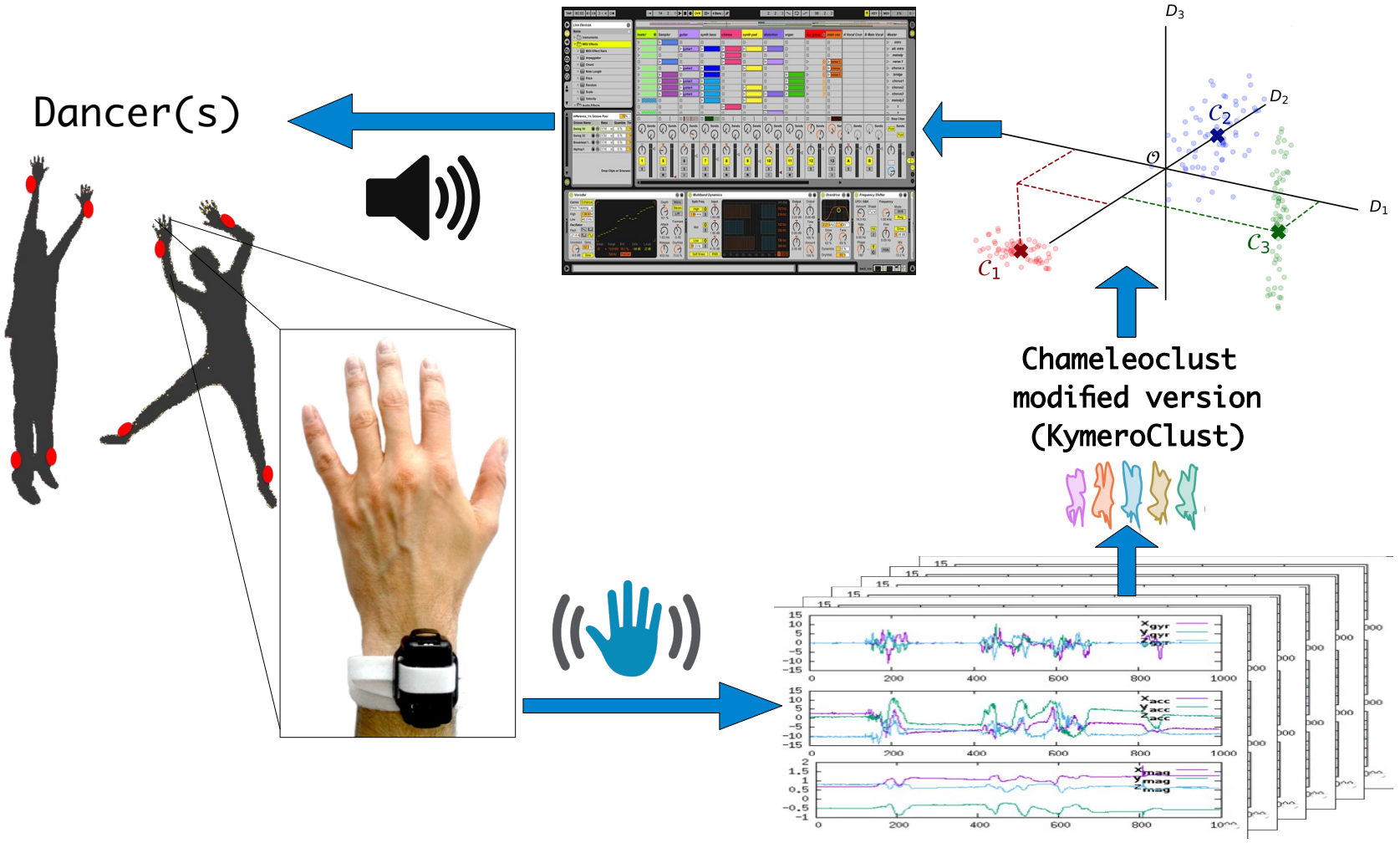




Chameleoclust
modified version
(KymeroClust)



EvoMove System.



EvoMove presentations.

- Tested by **professional dancers** (Anou Skan company)
- Presented in **8 performances in dance festivals**: *Meute* performance by the Désoblique dance company.
- **Promising feedback** from users (Qualitative appreciation).
  - Feel **real interaction** with the system.
  - Systems **reacts to changes** in the movements.

Outline

Conclusions

Hypothesis

It is possible to **take advantage** of an **evolvable genome structure** to tackle the **subspace clustering** task.

Conclusions

Hypothesis

It is possible to **take advantage** of an **evolvable genome structure** to tackle the **subspace clustering** task.

Conclusion

Incorporating **knowledge from evolution**:

- **Encode** different **clusters** in their **own subspaces**.
- **Adapt** to different datasets and dynamic data streams.
- Require **minor parameter tuning**.
- **Competitive results** with respect to the state-of-the-art techniques.

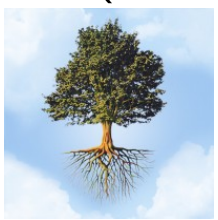
Perspectives

- **Apply**: Use **SubMorphoStream** with the **EvoMove** application.
- **Understand**: High degree of freedom $\rightarrow$ gains in terms of **quality** and **capacities to evolve**.
- **Explore**: Potential benefits of the **population** structure (ensemble clustering, temporality).

Acknowledgments

Supervisors:
Christophe Rigotti **Guillaume Beslon**

European project EvoEvo EU-FET (ICT- 610427) and partners.



LIRIS INRIA INSA CNRS

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.

$$\underbrace{o_1, o_2, \ldots, o_n}_{window}$$

- **Sliding** window.
- **Fading** window.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.

$$\underbrace{o_1, o_2, \ldots, o_n, o_{n+1}}_{window}$$

- **Sliding** window.
- **Fading** window.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.

$$\underbrace{o_1, o_2, \ldots, o_n, o_{n+1}, \ldots, o_N}_{window}$$

- **Sliding** window.
- **Fading** window.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.
- **Sliding** window.

$$\underbrace{o_1, o_2, \ldots, o_n}_{window}$$

- **Fading** window.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.
- **Sliding** window.

$$o_1, \underbrace{o_2, \ldots, o_n, o_{n+1}}_{window}$$

- **Fading** window.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.
- **Sliding** window.

$$o_1, o_2, \ldots, o_n, o_{n+1}, \underbrace{\ldots\ldots\ldots}_{window}, o_N$$

- **Fading** window.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

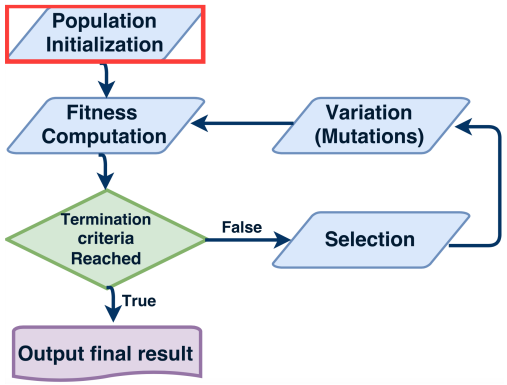
- **Landmark** window.
- **Sliding** window.
- **Fading** window.
  - At current time T objects are **weighted** according to their time stamps t (fade exponentially).
  - $weight(o_t) = 2^{-\lambda \times (T-t)}$, λ: **fading** parameter.
- **Tilted time** window.

Window models e.g. [Sim et al. DMKD 2013]

- **Landmark** window.
- **Sliding** window.
- **Fading** window.
- **Tilted time** window.
  - **Global picture** of the data stream.
  - **Fine granularity** for **recent** data and **coarse scale** for **old** ones.

$$\ldots, \underbrace{o_{n-32}, \ldots, o_{n-16}}, \ldots, \underbrace{o_{n-8}}, \ldots, o_{n-5}, \underbrace{o_{n-4}}, o_{n-3}, \underbrace{o_{n-2}}, \underbrace{o_{n-1}}, \underbrace{o_n}.$$

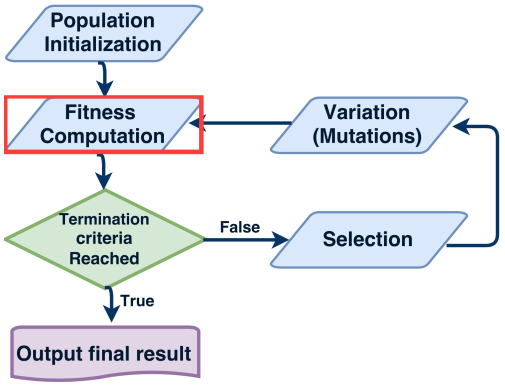
Characteristics of Clustering Evolutionary Algorithm [Hruschka et al. 2009]



Genome structure

- **Binary**, **Integer** and **Real** encoding.
- Cluster **memberships**, **medoids** or **centroids** are encoded.
- **Fixed** and **variable** number of clusters.

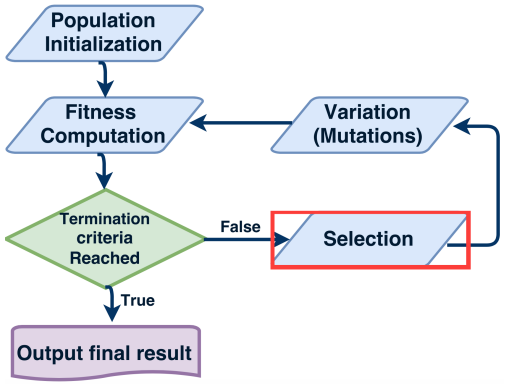
Characteristics of Clustering Evolutionary Algorithm [Hruschka et al. 2009]



Fitness functions

- Algorithms with **fixed** number of clusters:
  - Sum of **intra-cluster distances**.
  - **Clustering-Oriented** family.
- Algorithms with **variable** number of clusters:
  - Different **coefficients** (e.g., Silhouette Coefficient).
  - **Multi-objective** fitness function.

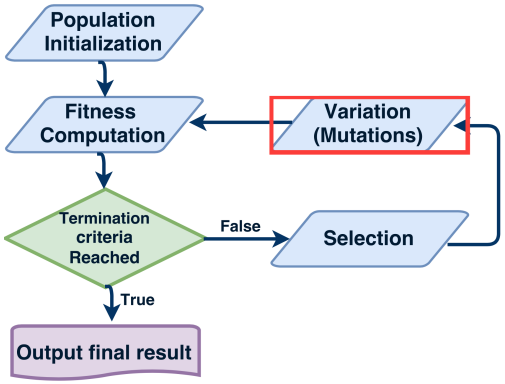
Characteristics of Clustering Evolutionary Algorithm [Hruschka et al. 2009]



Selection schemes

- **Proportional** selection scheme.
- **Elitist** variants.
- **Tournament** selection.
- $(\mu + \lambda)$-ES.

Characteristics of Clustering Evolutionary Algorithm [Hruschka et al. 2009]



Mutational operators

- **Cluster-oriented** and **non-oriented** operators.
- **Guided** and **not-guided** operators.

Evolutionary Algorithms applied to clustering

Nocea [Sarafis et al. 2007]

- **Cell-based** approach.
- Rule-based **integer** encoding.
- **Variable** number of clusters.
- Subspaces produced **a-posteriori**.

S-ESC [Vahdat et al. 2013]

- **Density-based** and **Clustering-oriented** approach.
- **Multiobjective** optimization.
- Two populations.
- Rely on a first **non-evolutionary stage**.

Scalable-ECSAGO [León et al. 2010]
ESCALIER [Veloza et al. 2013]

- **Clustering-oriented** family.
- **Sliding window**.
- **Variable** number of clusters.

- **No subspace clustering of data streams based on Evolutionary Algorithms.**

Fitness computation
Assignment Mismatch

- $\mathcal{S}^t$ set of **normalized data objects** observed at **generation** t.
- **Fitness** at **generation** t over a **sample** $\mathcal{S}^t \subseteq \mathcal{S}$
- **Assign objects** in $\mathcal{S}^t$ to **core-points** in phenotype Φ.
- $\underbrace{o_1, o_2, \ldots\ldots}_{\mathcal{S}^1} \underbrace{\ldots\ldots\ldots, o_{|\mathcal{S}|}}_{\mathcal{S}^2}, \underbrace{o_1, \ldots\ldots, x_{|\mathcal{L}|}}_{\mathcal{S}^t}, \ldots$

Main features

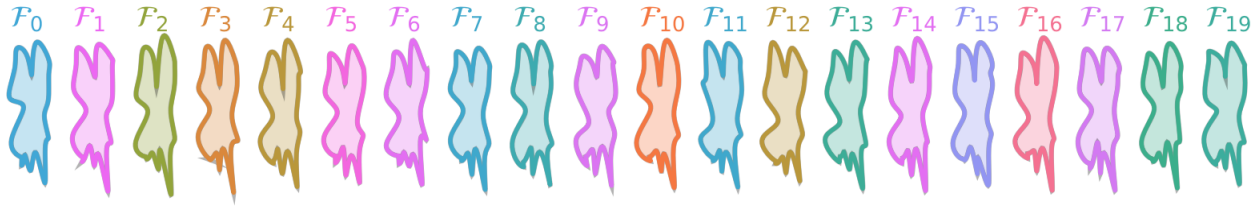
- **Extract core principles** from Chameleoclust
- **Abstract** genotype-phenotype joint **representation**.
- **Simple bio-inspired operators** (genes duplication/divergence) using data objects

Results summary

- **Competitive** w.r.t. state-of-the-art algorithms.
- Better results in **shorter runtimes**.

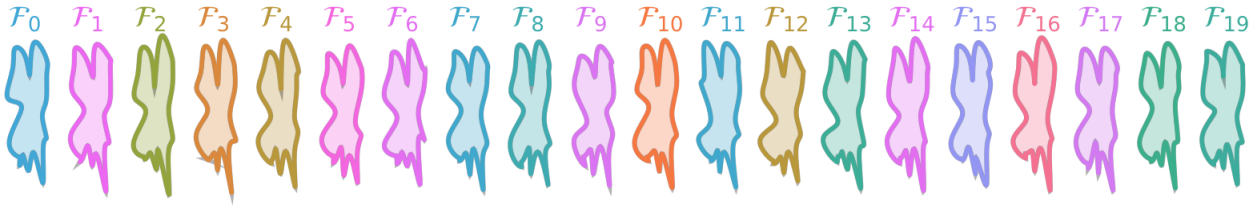
Selection

- Generation $\mathbf{t}$:

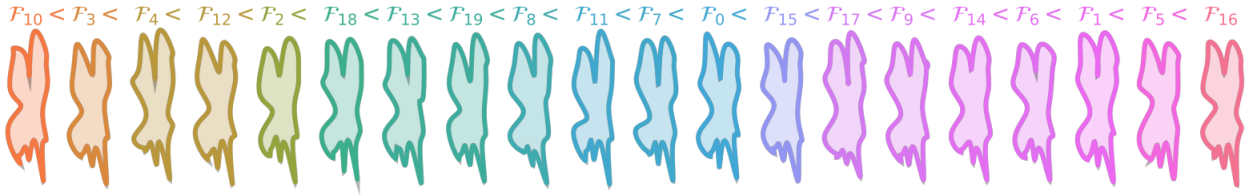


Selection

- Generation **t**:

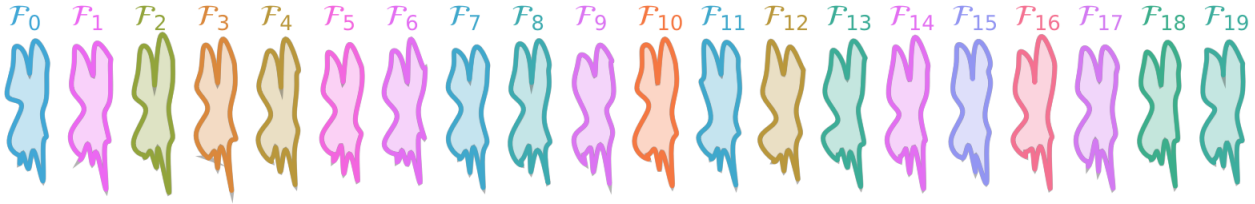


- **Ordering individuals by fitness** (Highest rank $\rightarrow$ best)

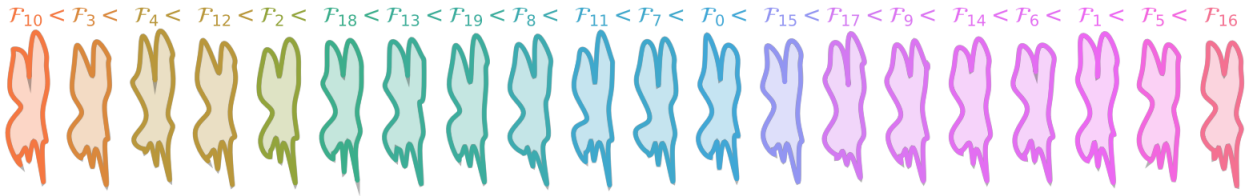


Selection

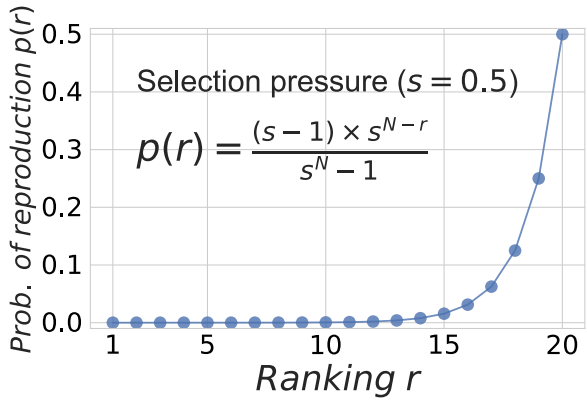
- Generation **t**:



- **Ordering individuals by fitness** (Highest rank $\rightarrow$ best)

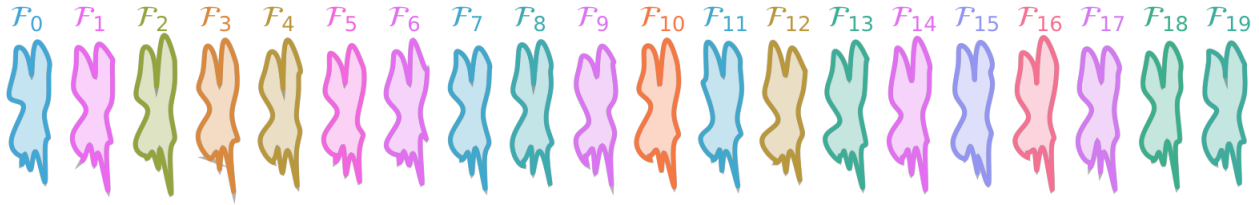


- **Exponential ranking** selection:

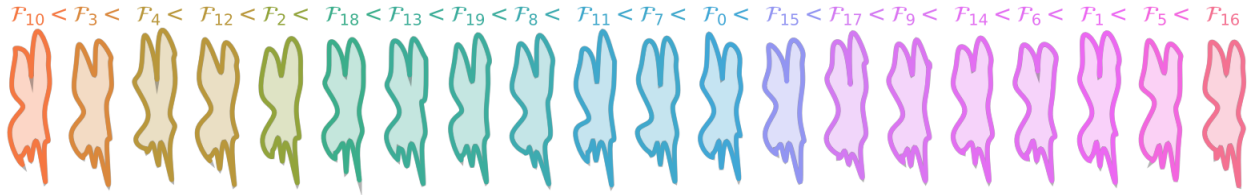


Selection

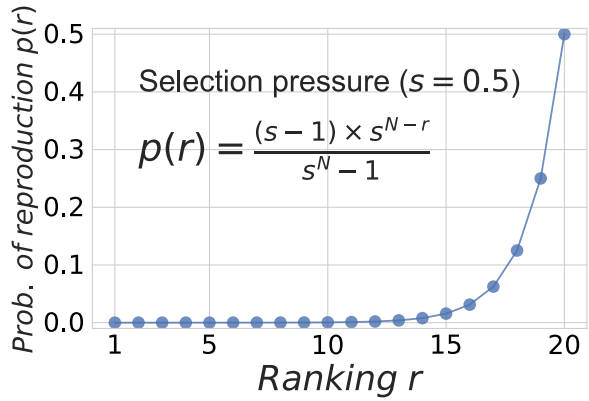
- Generation **t**:

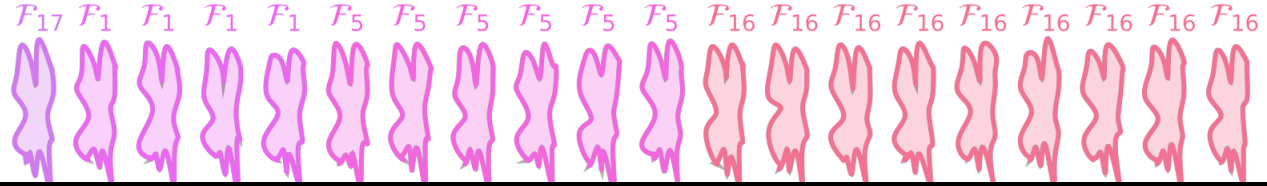


- **Ordering individuals by fitness** (Highest rank $\rightarrow$ best)



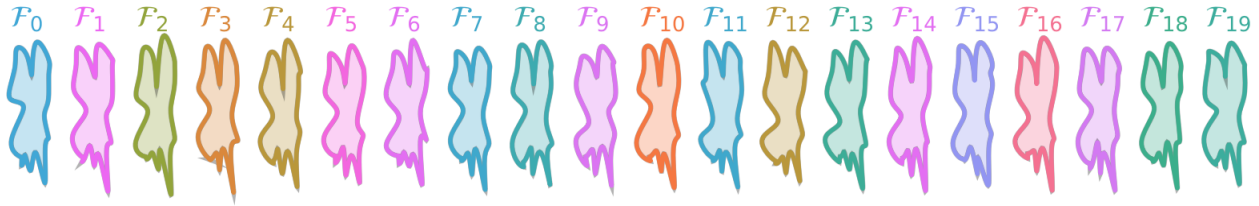
- **Exponential ranking** selection:



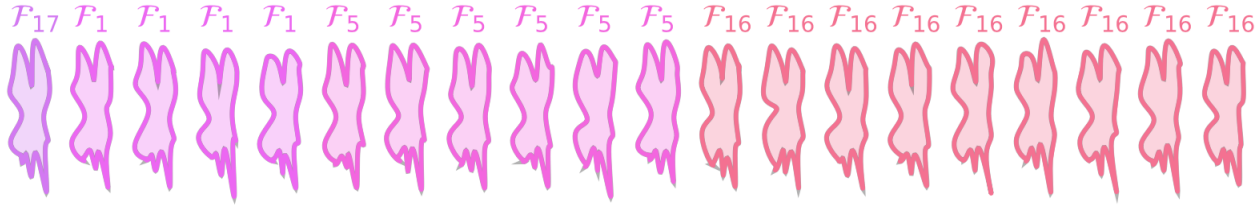


Selection

- Generation $\mathbf{t}$:

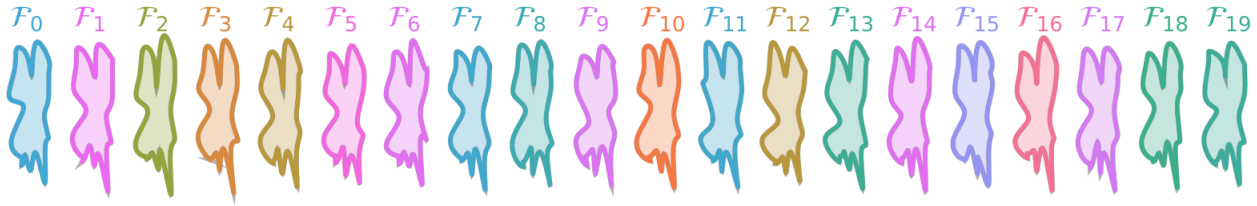


- Generation $\mathbf{t+1}$:

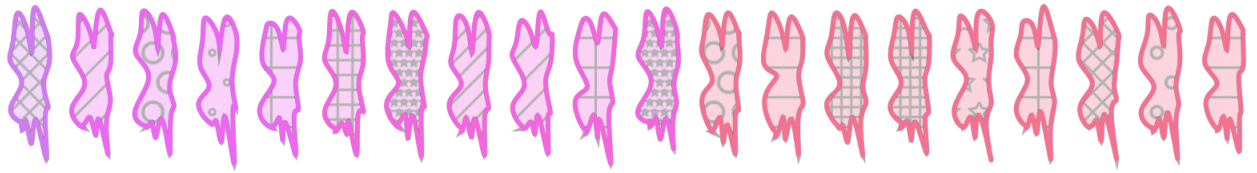


Selection

- Generation $\mathbf{t}$:

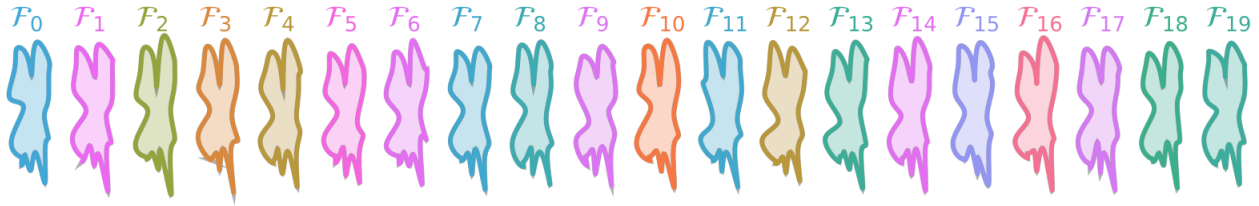


- Generation $\mathbf{t+1}$:

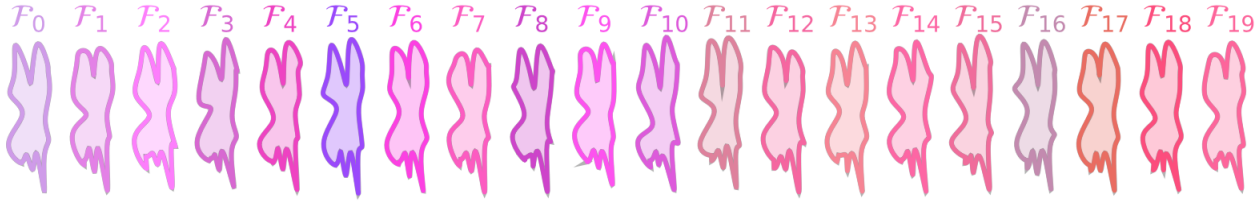


Selection

- Generation $\mathbf{t}$:

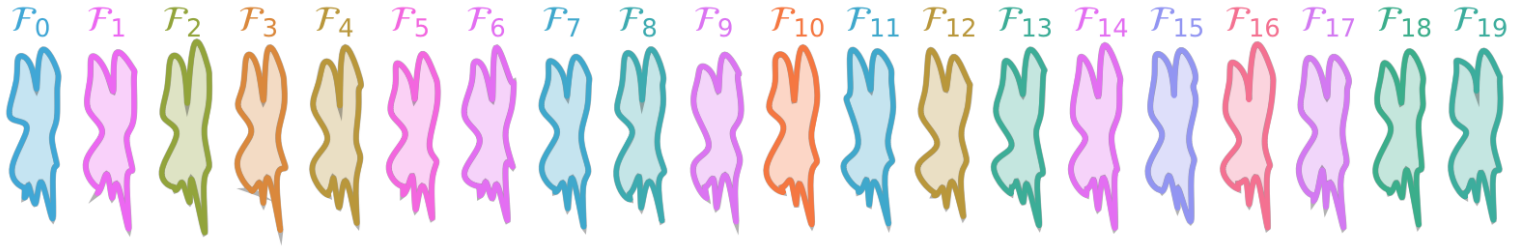


- Generation $\mathbf{t+1}$:



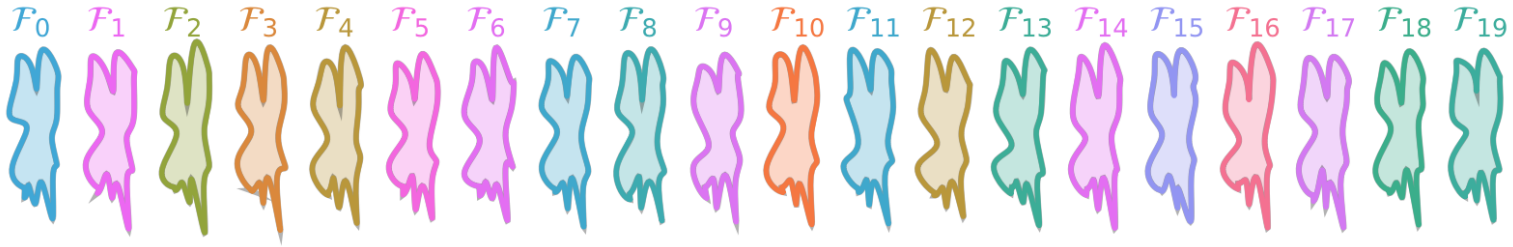
Selection

- Generation $\mathbf{t}$:



Selection

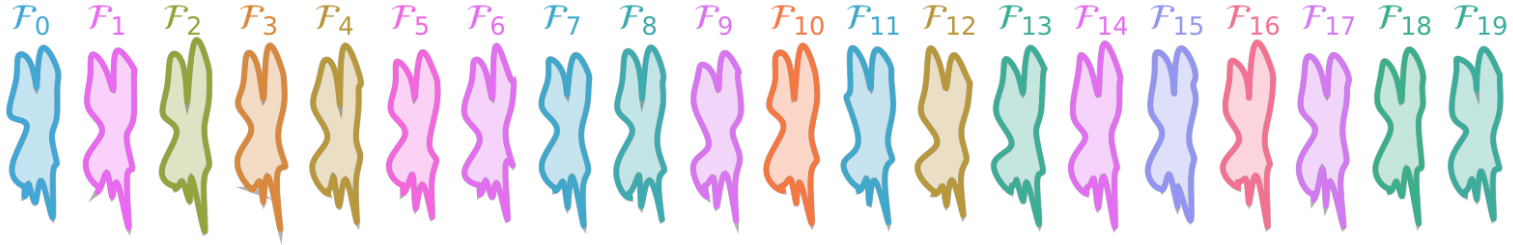
- Generation **t**:



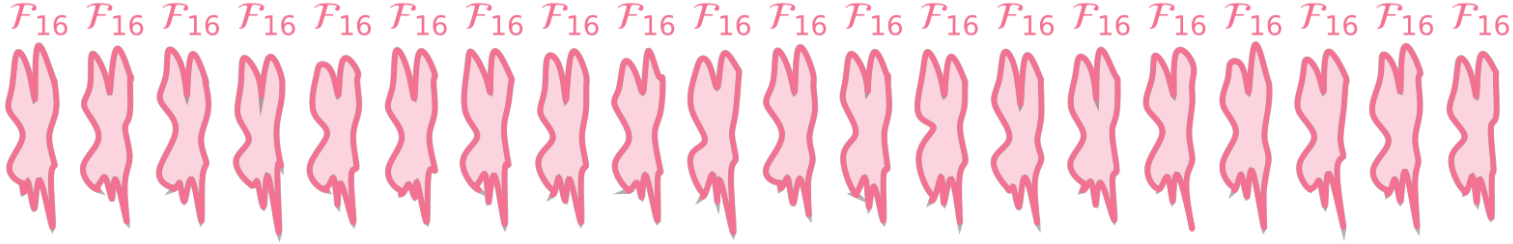
- $(1,\lambda)$-ES selection scheme:

Selection

- Generation **t**:

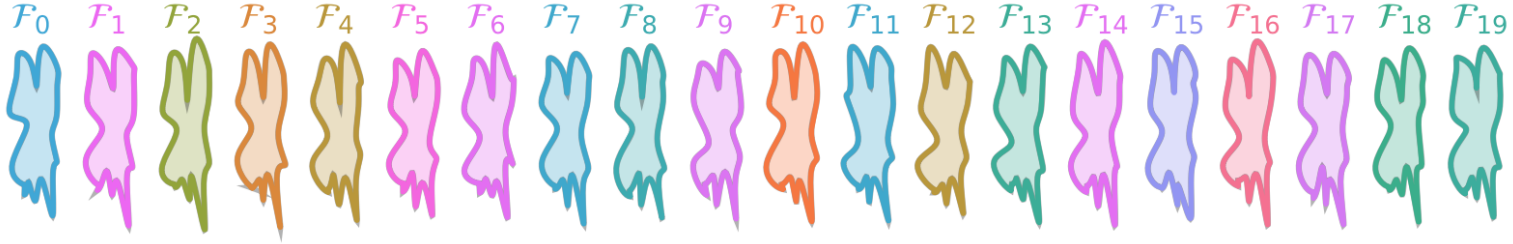


- Generation **t+1**:

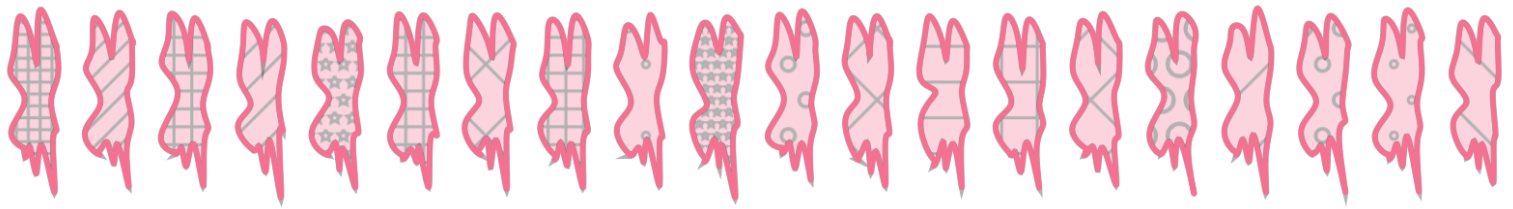


Selection

- Generation **t**:

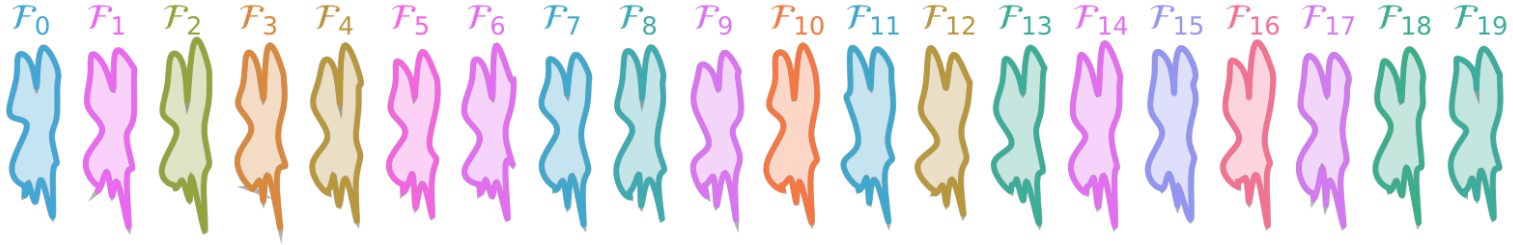


- Generation **t+1**:

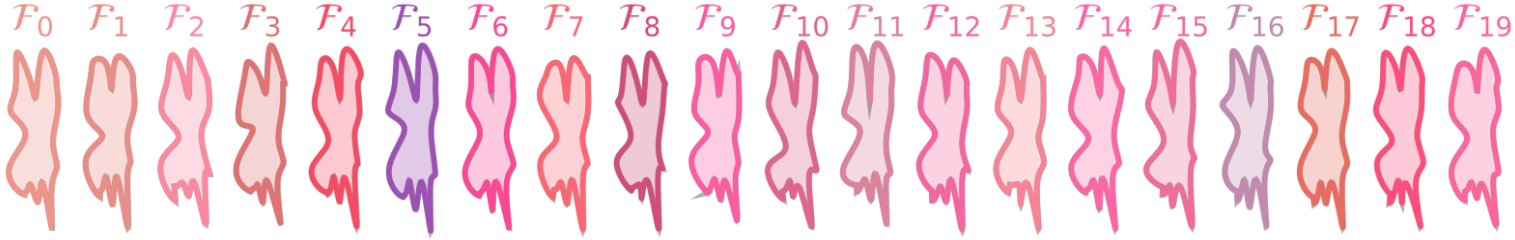


Selection

- Generation $\mathbf{t}$:



- Generation $\mathbf{t+1}$:



Time complexity Chameleoclust

- N: Number of individuals.
- D: Dimensionality.
- ω: Number of objects in the sample.
- $|\Gamma|$: Genome size.
- L_m: Maximal genome size reached during the rearrangement step.

Fitness computation complexity

$$\mathcal{O}(N \times |\Gamma| \times (D \times \omega + ln(|\Gamma|)))$$

Reproduction operations complexity

$$\mathcal{O}(N \times (ln(N) + |\Gamma| + |\Gamma| \times L_m + L_m))$$

Time complexity KymeroClust

- λ: Number of children.
- D_{max}: Dimensionality of the dataset.
- *NbCenters*: Number of core points.
- ω: Number of objects in the sample.

Complexity

$$\mathcal{O}(\lambda \times \omega \times NbCenters \times D_{max})$$

Time complexity SubMorphoStream

- K: Number of core-points.
- D: Dimensionality.
- λ: Number of children.
- η : Number of evolution generations per data object.
- ω: Genome size.

Reproduction operations complexity

- Exogenous genetic uptake: $\mathcal{O}(\lambda \times \eta \times (K + D))$
- Amplification: $\mathcal{O}(\lambda \times \eta \times D \times K)$

Fitness computation complexity

$$\mathcal{O}(\lambda \times \eta \times \omega \times D \times K)$$

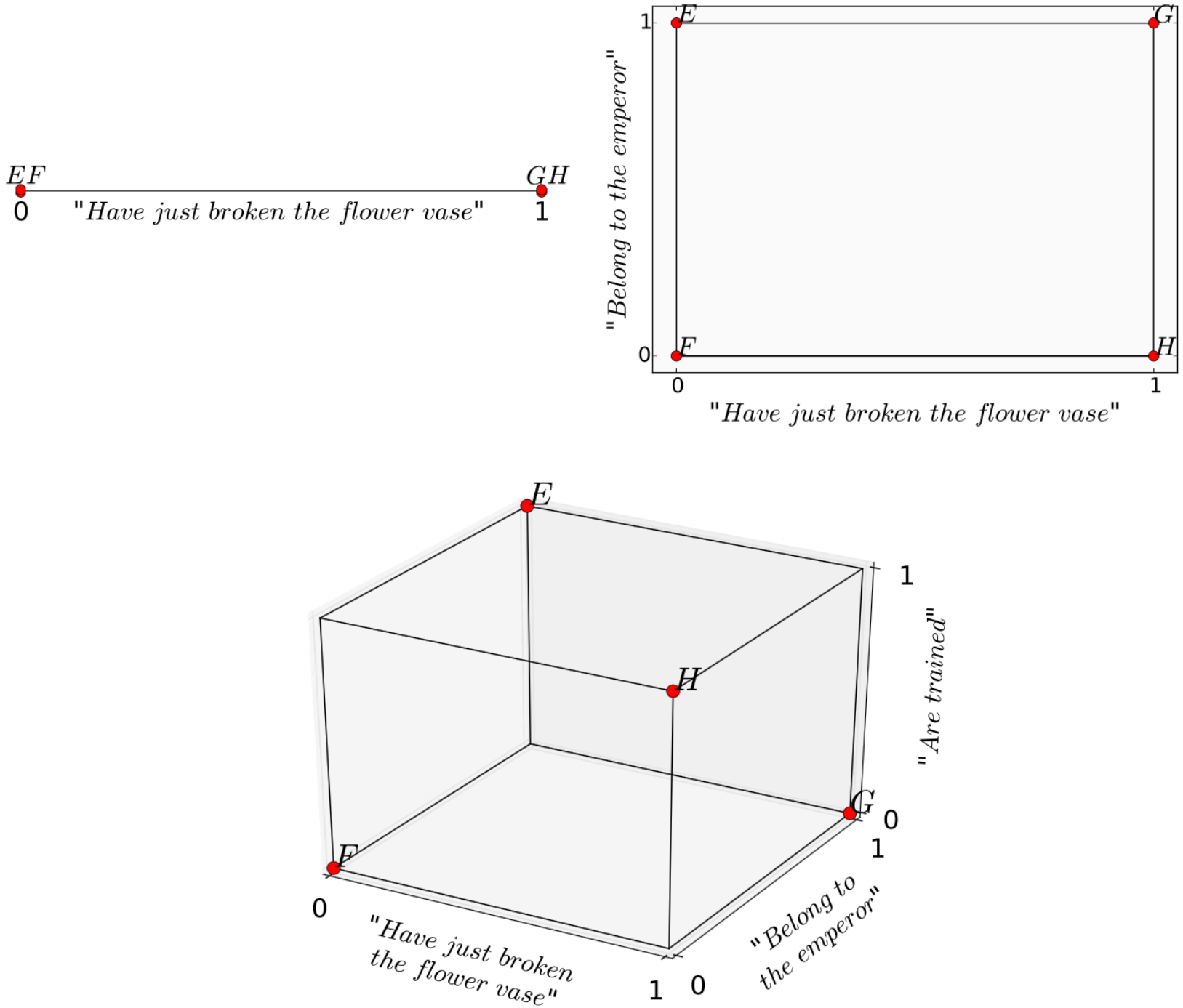


FIGURE 1.1: Toy example to illustrate the *curse of dimensionality*