Framsticks model and genetics

- organism model
  - body
  - brain
  - sensors and effectors
- genetics
  - representations
  - conversions
  - operators

Organism. Elements

- body
  - parts
  - joints
- brain
  - neurons
    - signal processing / sensors / actuators
    - embodied or not
  - connections

Organism. Body elements

- Parts
  - **type: ellipsoid / cylinder / box**
  - size: x, y, z
  - 3D position
  - 3D orientation
  - mass
  - friction
  - (ingestion, assimilation, ...)
- Joints
  - references of the two parts
  - relative or not
  - stiffness
  - rotation stiffness
  - (stamina, ...)

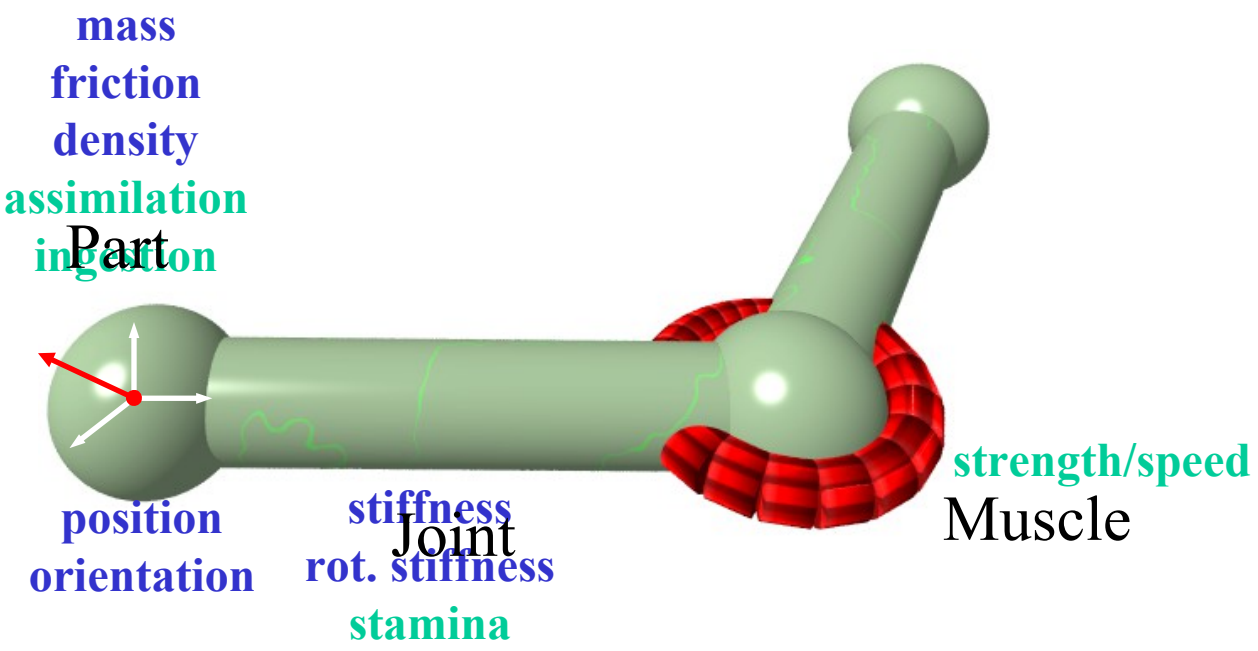
Organism. Body constraints

- at most one Joint can directly connect two Parts
- each Joint must be connected with two distinct Parts
- all Parts must be directly or indirectly connected with each other
- relative Joints must not form cycles

Organism. Body properties

PHYSICAL	BIOLOGICAL
Parts: mass, friction, density	Parts: assimilation, ingestion
Joints: stiffness, rotational stiffness	Joints: stamina
	Muscles: strength/speed

CONSTRAINTS



Organism. Brain

- any topology of neural network, synchronous update
- neurons embodied (parts, joints) or not
- implement any function
- inputs: none / one / many
- outputs: none / one (may have many channels)
- a list of neural properties (parameters)
- definition: C++ or script (*.neuro files)
- weighted connections

Organism. Brain. Neuron sample

Short name: Thr

Long name: Threshold

- single input
- single output
- properties:
  - t (threshold)
  - hi (high output value)
  - lo (low output value)
- if (input>=t) then output:=hi else output:=lo

Organism. Brain. Neuron sample

Short name: N

Long name: Neuron

- many inputs
- single output
- properties:
  - fo (force)
  - in (intertia)
  - si (sigmoid)

$$o_t = \frac{2}{1 + e^{-s_t \cdot Sigmoid}} - 1$$

$$s_t = s_{t-1} + v_t$$

$$v_t = v_{t-1} \cdot inertia + force \cdot (i_t - s_{t-1})$$

i – weighted sum of inputs
v – speed of changes
s – internal state
o – neuron output

subscript t is the moment of time

$$force := 1$$

$$inertia := 0$$

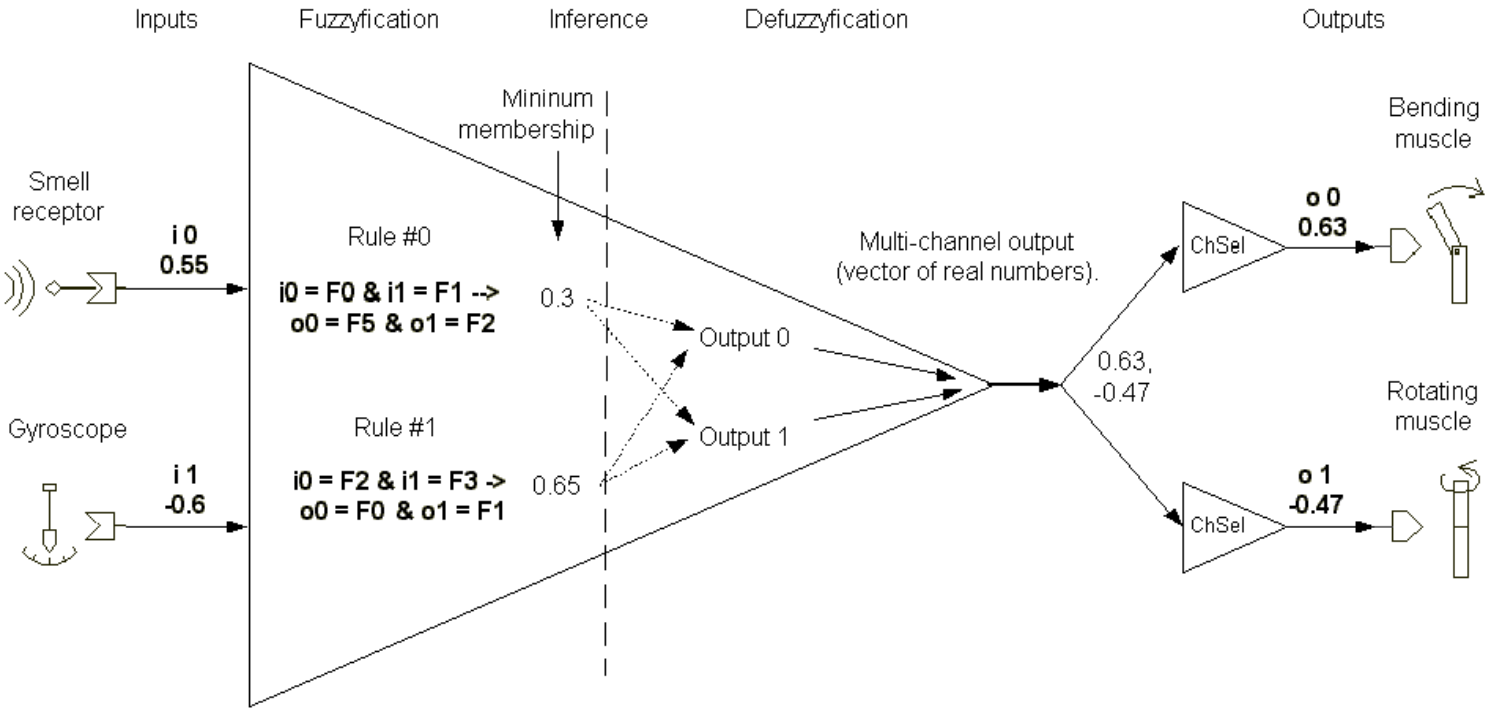
$$o_t = \frac{2}{1 + e^{-i_t \cdot Sigmoid}} - 1$$

Organism. Brain. Neuron sample

Short name: Fuzzy

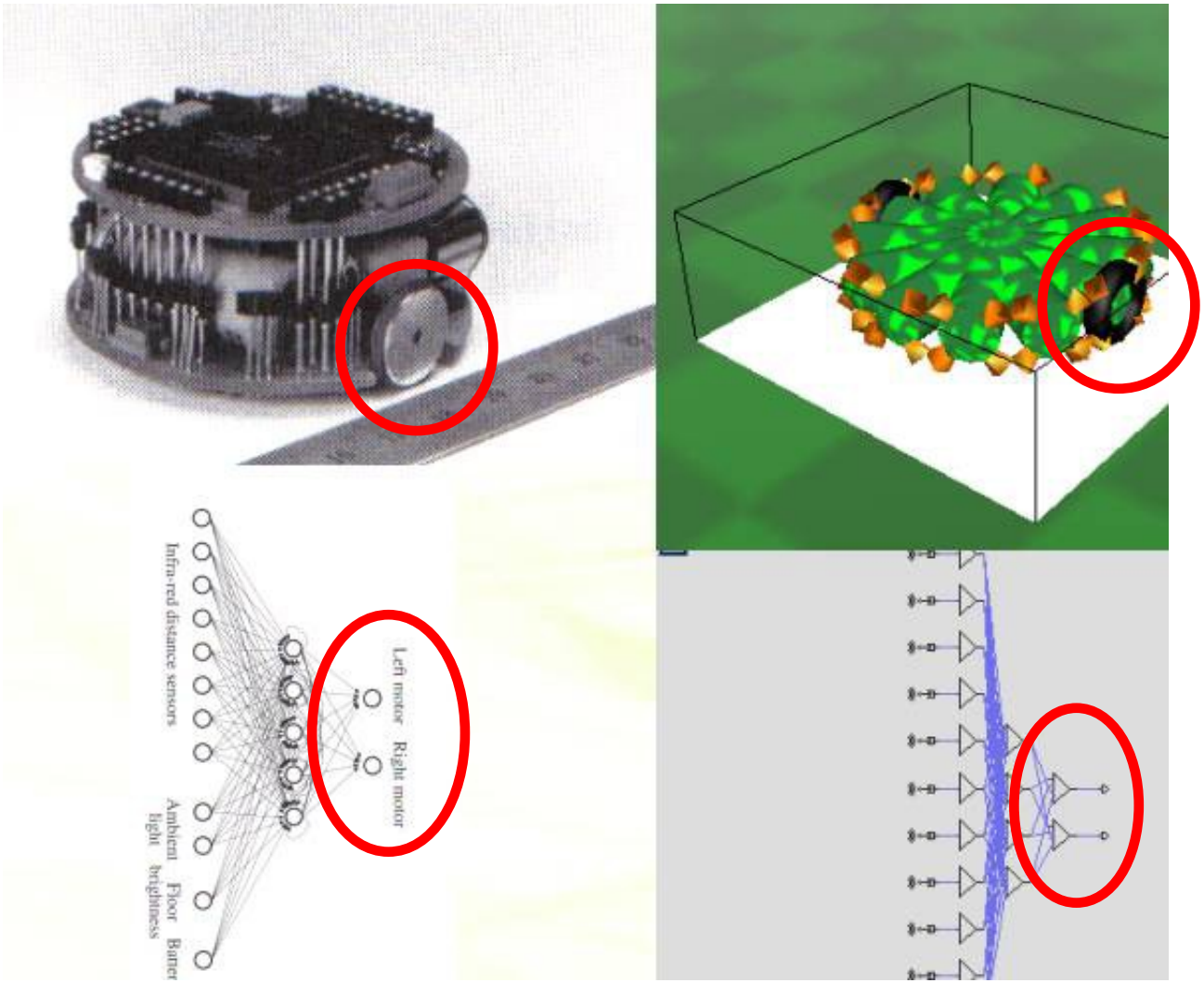
Long name: Fuzzy neuron

- many inputs
- single output (with many channels)
- properties: fuzzy sets and rules
- represents a fuzzy rule-based system
- details: M. Hapke and M. Komosinski. Evolutionary design of interpretable fuzzy controllers. *Foundations of Computing and Decision Sciences* 33(4), 2008. [view pdf]



Organism. Brain. Neuron sample

A custom ”Wheel” effector for robotic experiments

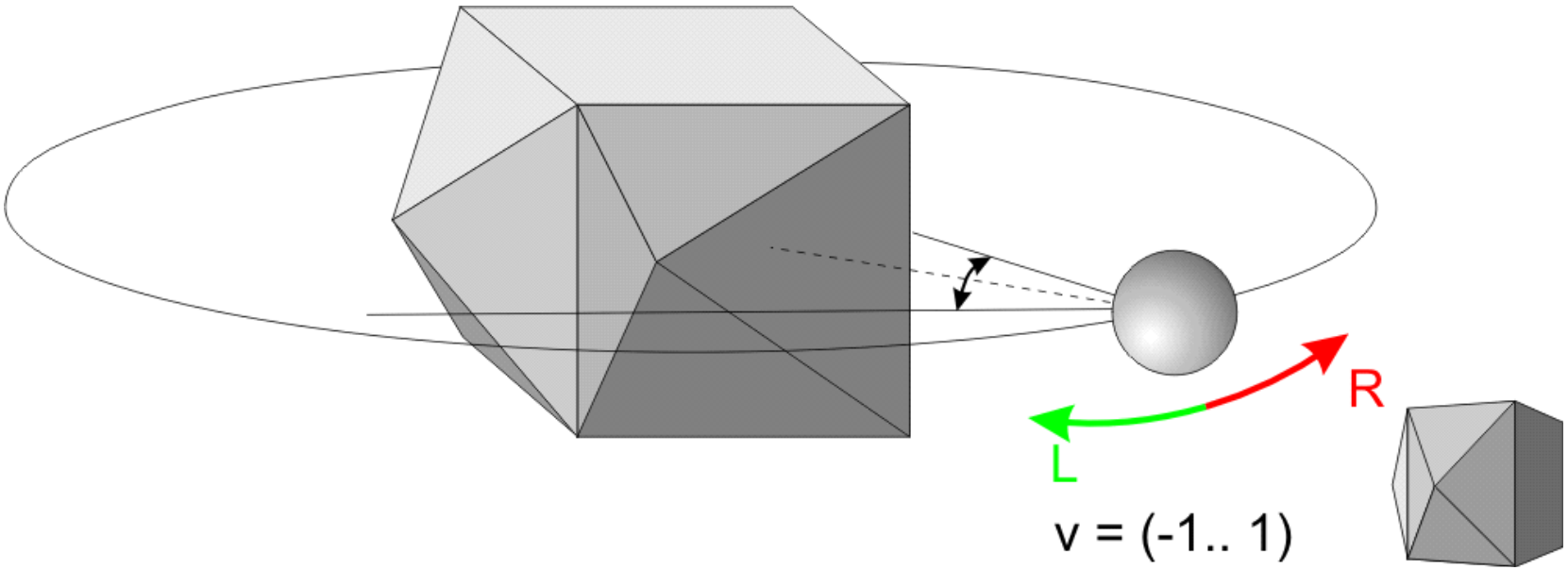


- affects movement of a Part of the creature

Organism. Brain. Neuron sample

A vector eye (VEye) sensor

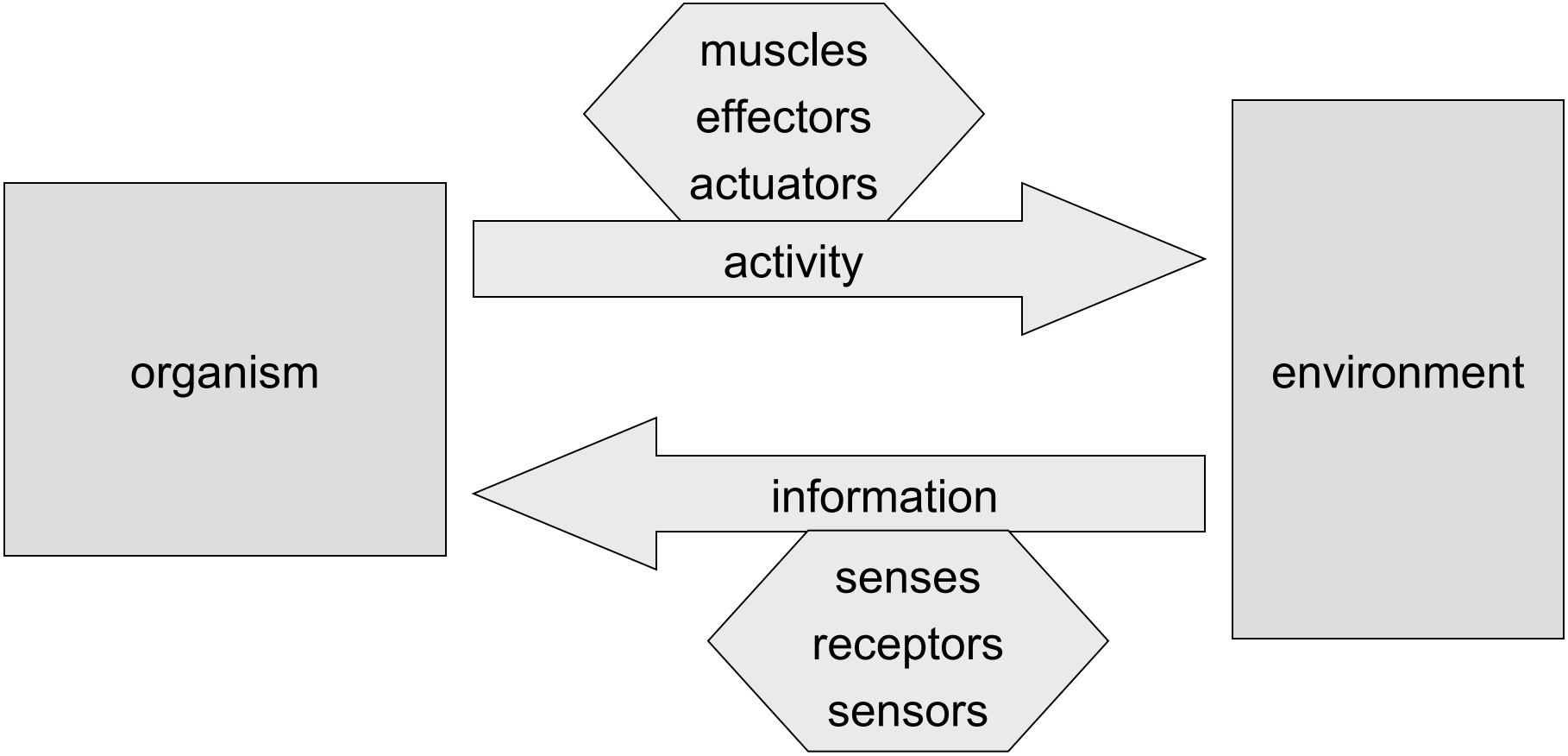
- optional input controls tilt (rotation)
- single output (with many channels) outputs vector coordinates
- properties: object (sic!), scale, perspective
- details: Jacek Jelonek and Maciej Komosinski. Biologically-inspired visual-motor coordination model in a navigation problem. In Knowledge-Based Intelligent Information and Engineering Systems, LNCS vol. 4253, pages 341-348. Springer, 2006. [view pdf]



Organism. Brain. Neuron list

N	Neuron	Standard Framsticks neuron
G	Gyroscope	Equilibrium sensor
T	Touch	Touch sensor
S	Smell	Smell sensor
*	Const	Constant value
|	Bend muscle	
@	Rotation muscle	
D	Differentiate	Calculate the difference between the current and previous input value
Ch	Channelize	Combines all input signals into single multichannel output
ChMux	Channel multiplexer	Outputs one channel from first (multichannel) first input, selected by the second
ChSel	Channel selector	Output one channel from multichannel input, selected by the "ch" parameter
Rnd	Random value	
Sin	Sinus Generator	Output frequency = f0+input
Delay	Delay	
Thr	Threshold	if (input>=t) then output=hi else output=lo
Fuzzy	Fuzzy neuron	
VEye	Vector eye	
LMu	Length muscle	
Water	Water detector	
Energy	Energy level	

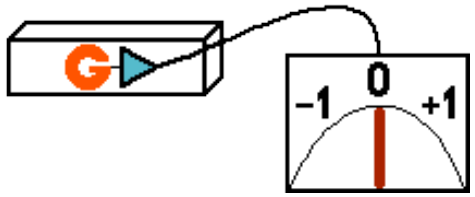
Organism. Simulation. Interactions



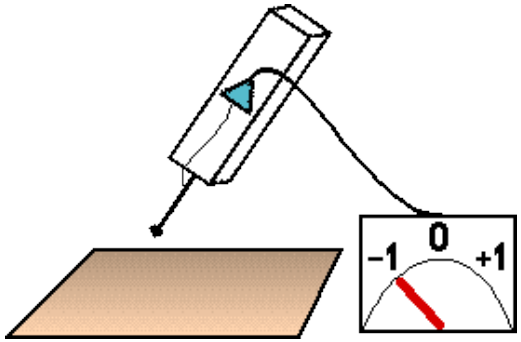
Organism. Body/Brain

Receptors and effectors

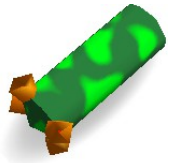
equilibrium

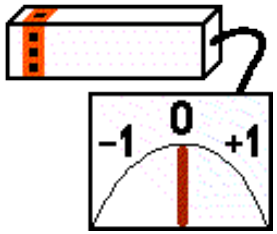


touch

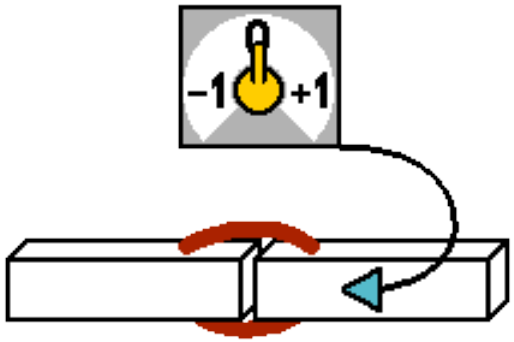


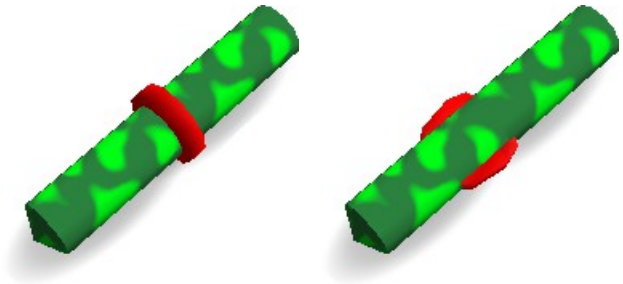
smell



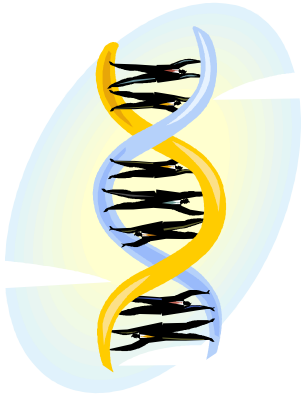


bending and rotating muscles





Genetics. Transformations.

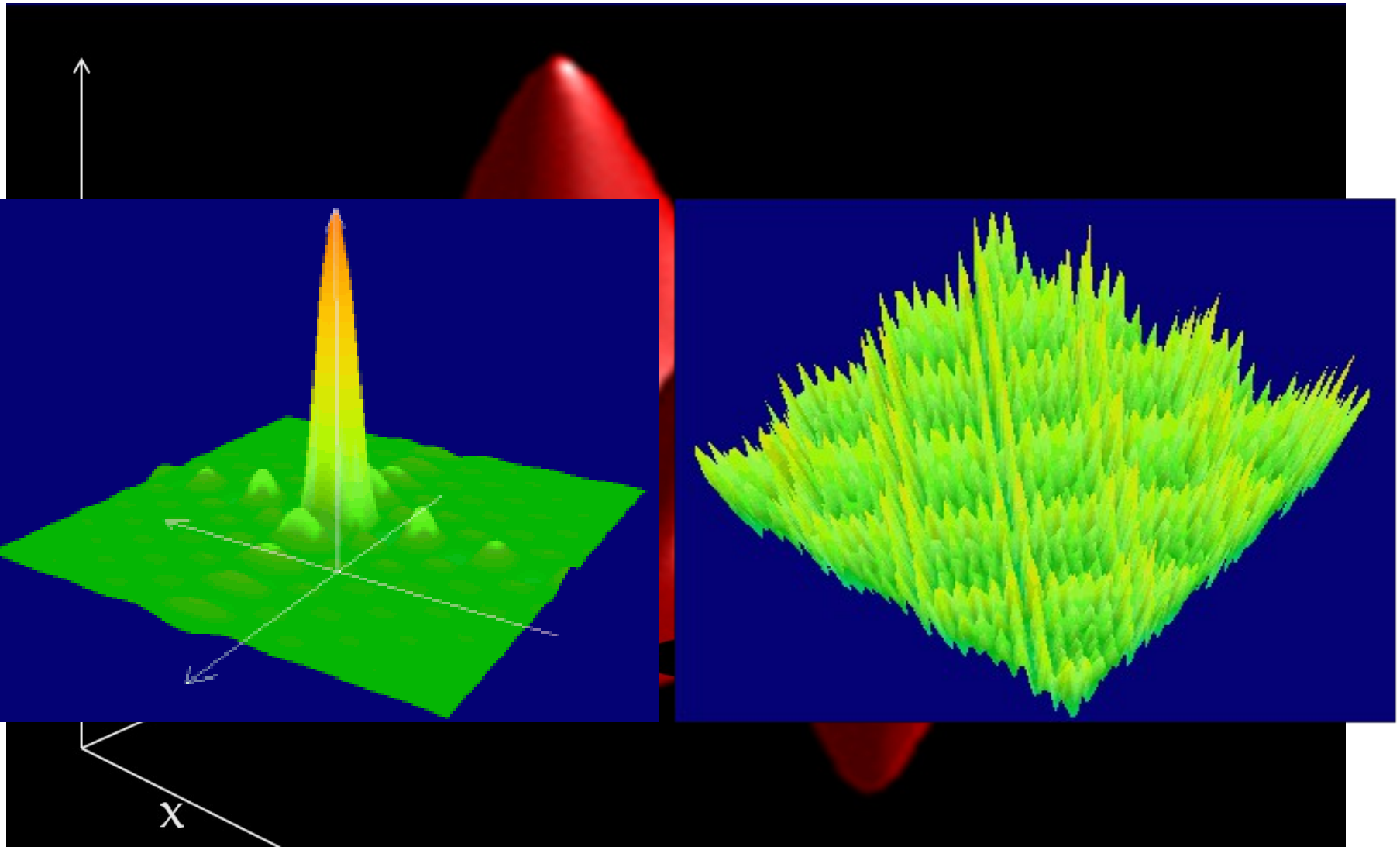


DNA ⟶ organism 

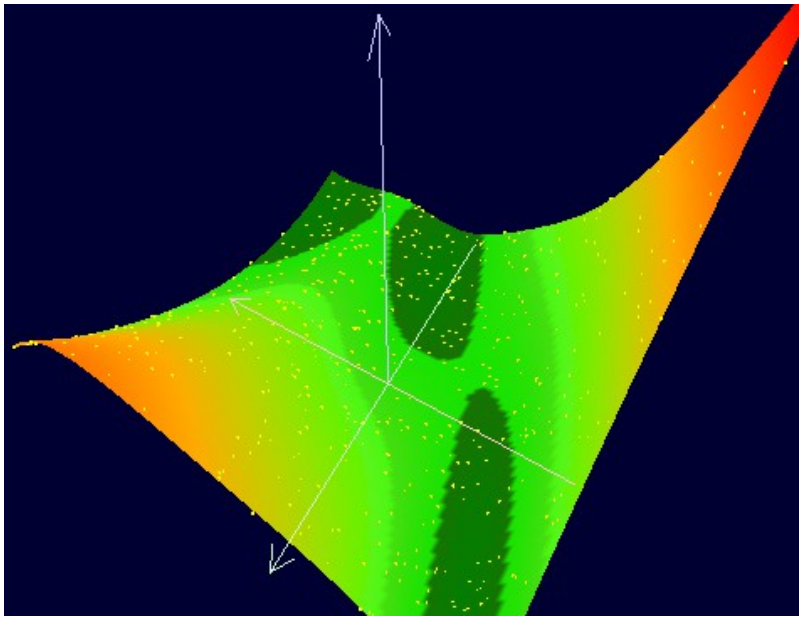
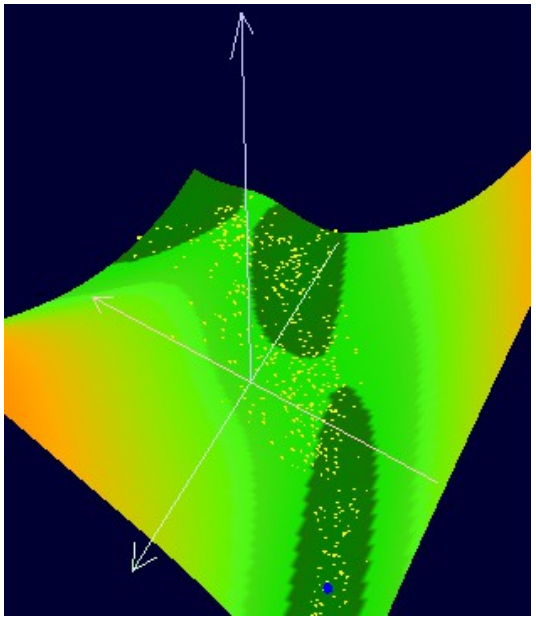
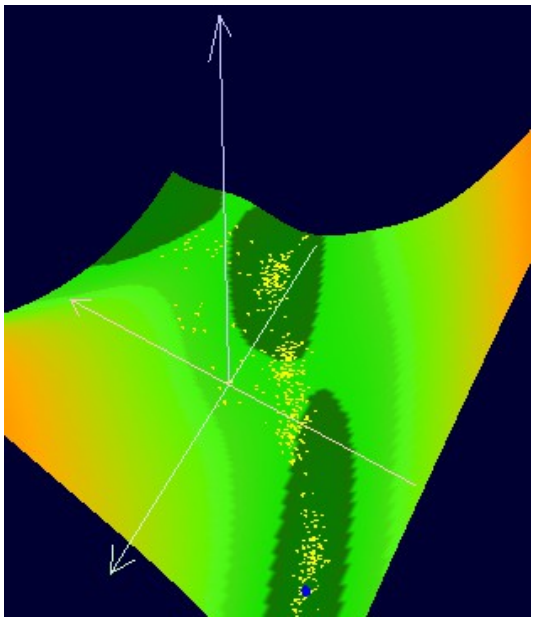
genotype ⇉ model

Why so important?

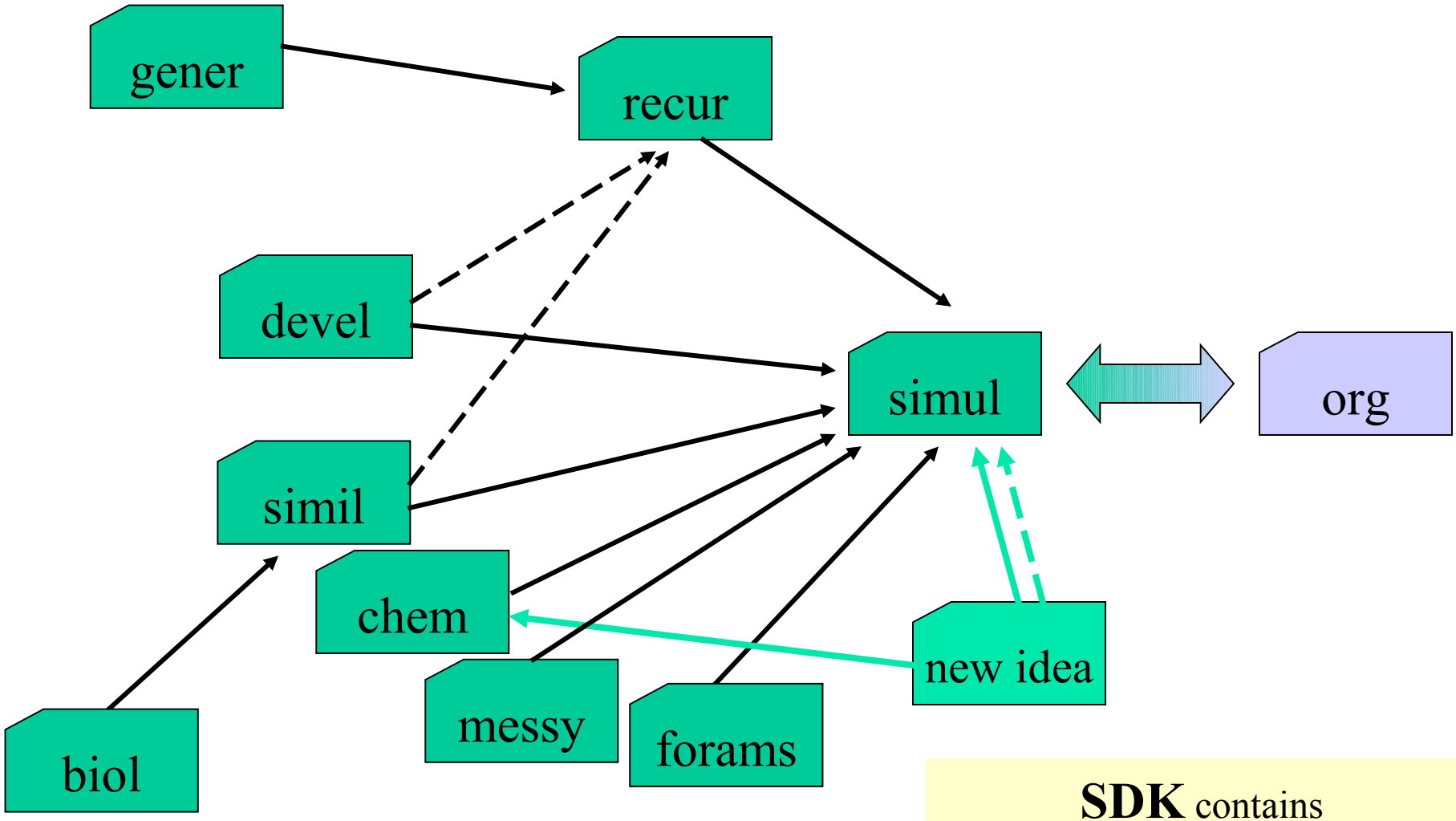
Fitness landscapes!



Evolutionary algorithms

Representation conversion graph

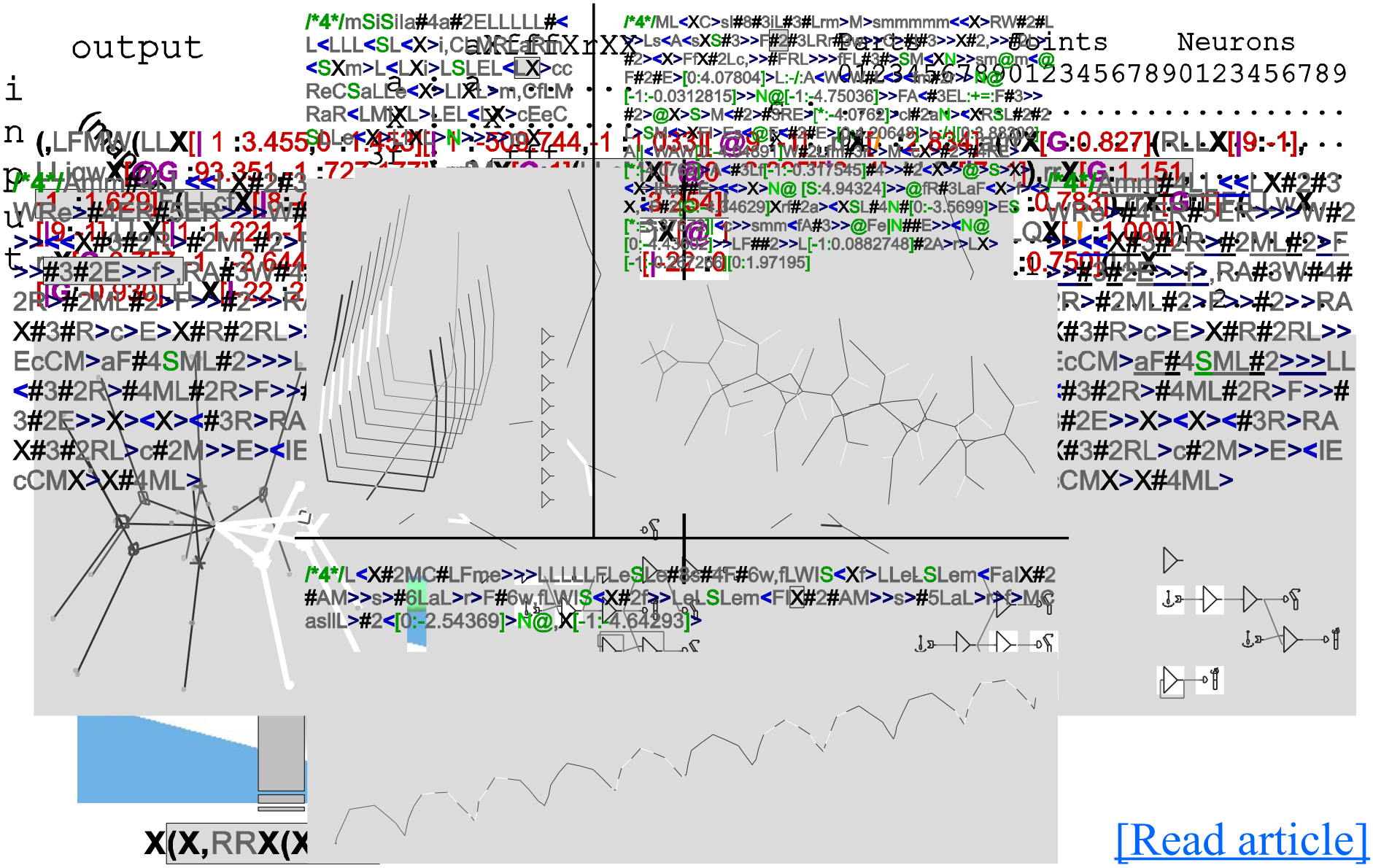


SDK contains
"Genotype Developer Kit"

Framsticks genetics

Genetics. Mapping.

gene ⇔ gene ⇔... ⇔ phene



[Read article]

Characteristics of genetic representations

[Read article]

	Complexity		Constraints	
	Genotype	Interpretation	Body	Brain
simul	Med	Low	None	None
recur	Med	Med	High	Low
devel	High	Med	High	Low

	Modularity	Symmetry	Compression	Redundancy
simul	None	None	None	None
recur	None	Low	None	Low
devel	High	High	Var	None

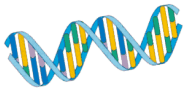
Characteristics of genetic representations

	Complexity		*Constraints*		*Cyclic Body*	*Compression*	*Redundancy*
	Format	*Interpret*	*Body*	*Brain*			
simul	Med	Low	None	None	Y	None	None
recur	Med	Med	High	Low	N	None	Low
simil	Low	High	Med	None	Y/N	Low	None
chem	Low	High	Med	None	Y/N	Var	Var
devel	High	Med	High	Low	N	Var	None
messy	High	Low	High	?	N	None	None
...							

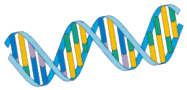
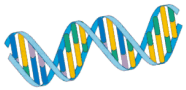
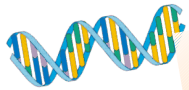
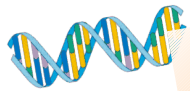
Genetic operators

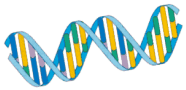
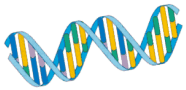
format

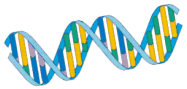
12.574
12.574
12.574
12.574

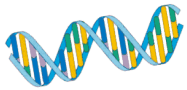
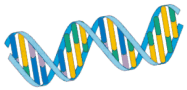

Mutation

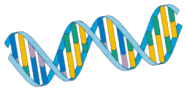
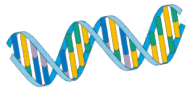
12.574



Crossover

12.574

12.574


Repair



Find error
12.574



Estimate similarity
12.574


Simplify


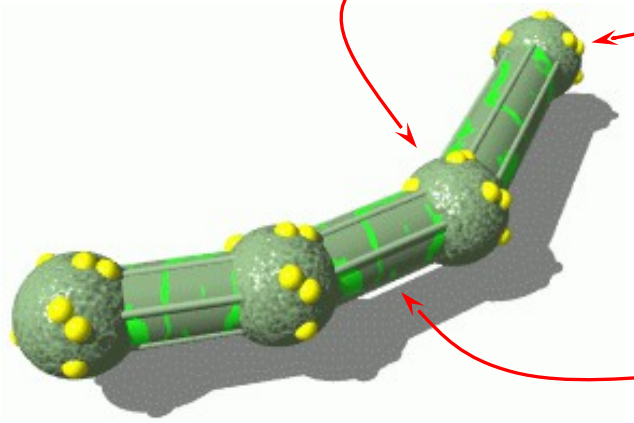
Simul representation

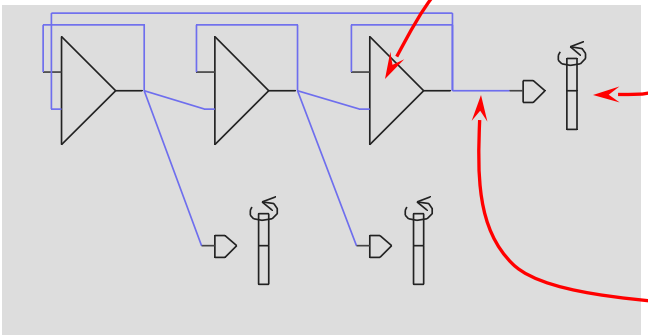
- all parts directly described
- basic, internal format
- supports geometric relativity

qX(X[@,1:1],X[Sin])

```
p:
p:1, m=3
p:1.50017, -0.865927
p:1.50017, 0.865927
j:0, 1, dx=1
j:1, 2, rx=-0.62568, rz=-1.047, dx=1
j:1, 3, rx=-0.62568, rz=1.047, dx=1
n:j=1, d=@
n:p=3, d=Sin
c:0, 1
```

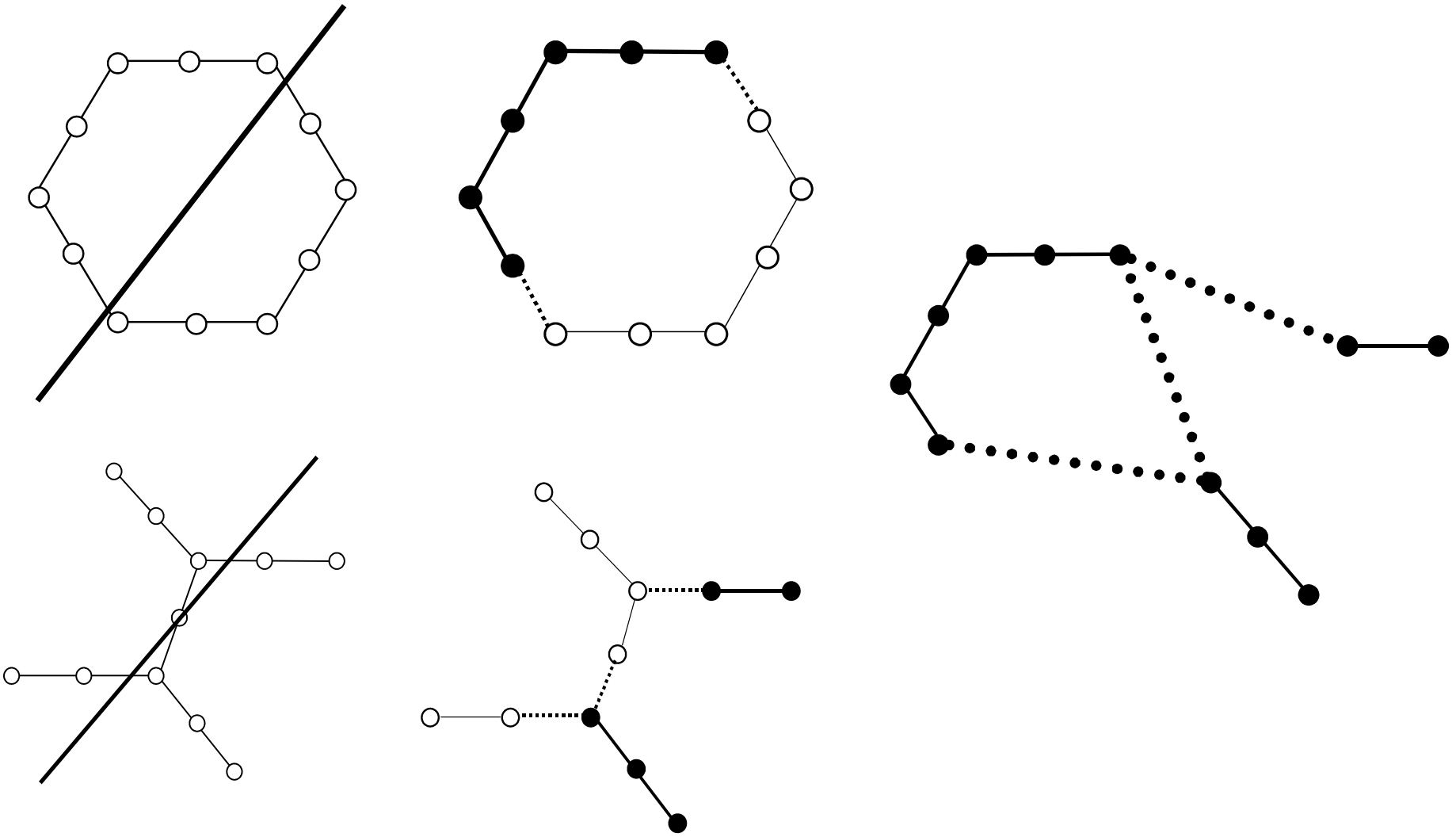
Simul genotype-phenotype relation



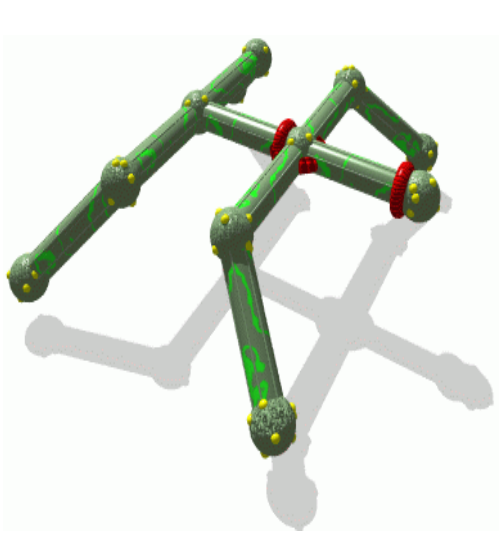


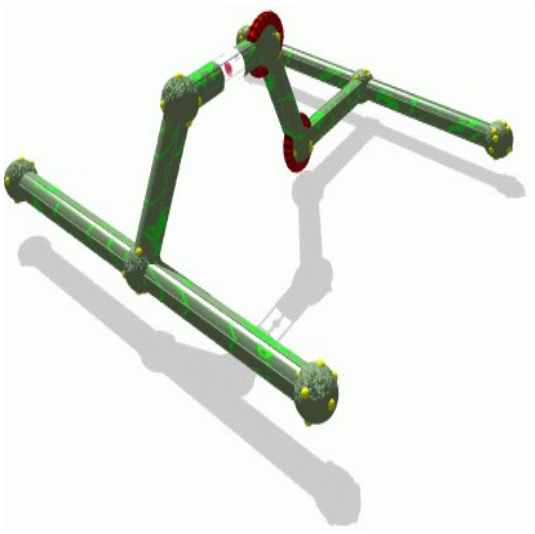
```
//0
parts
p:m=3
p:0.000145457, -0.490867
p:1, m=4, ing=0.0304878, as=0.0304878
p:1.00015, -0.339942, 0.354104, ing=0.0743, as=0.0743
...
joints
j:0, 1, rx=-0.8058, rz=-1.5705, dx=0.490867
j:0, 2, rx=-0.8058, dx=1, stam=0.0304878
j:2, 3, rx=-1.06366, rz=-1.5705, dx=0.49086, stam=0.07439
j:2, 4, rx=-1.06366, dx=1, stam=0.0236728
...
neurons
n:p=2, d=N:si=1.879
n:j=1, d=@:p=0.908537
n:p=4, d=N:si=-3.35
...
conn's
c:0, 0, 2.408
c:0, 2, -0.812
c:1, 0
```

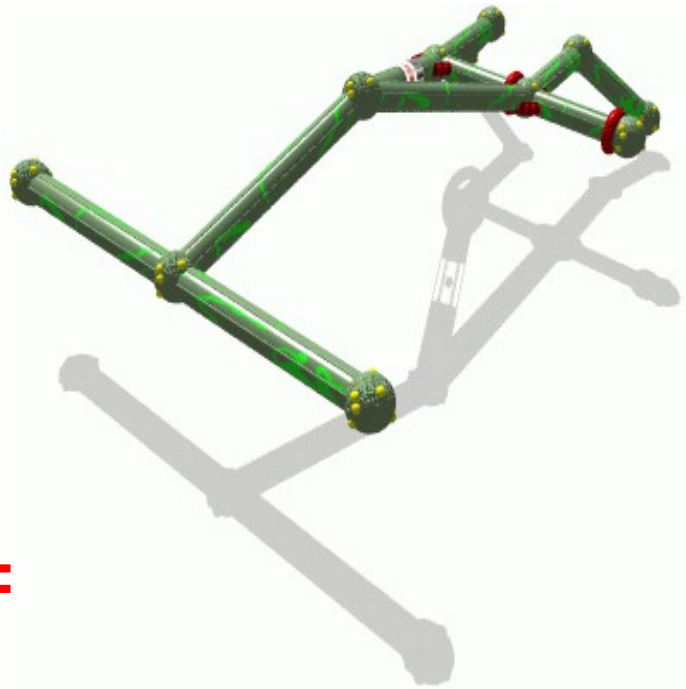
Simul crossing-over

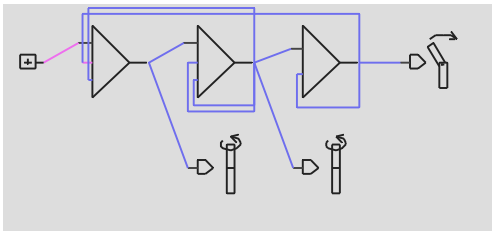


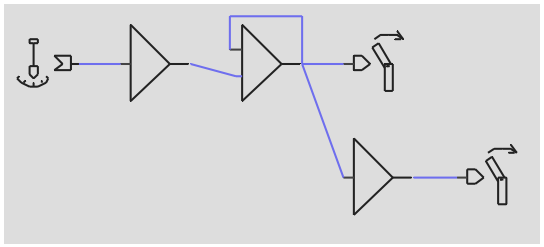
Simul crossing-over

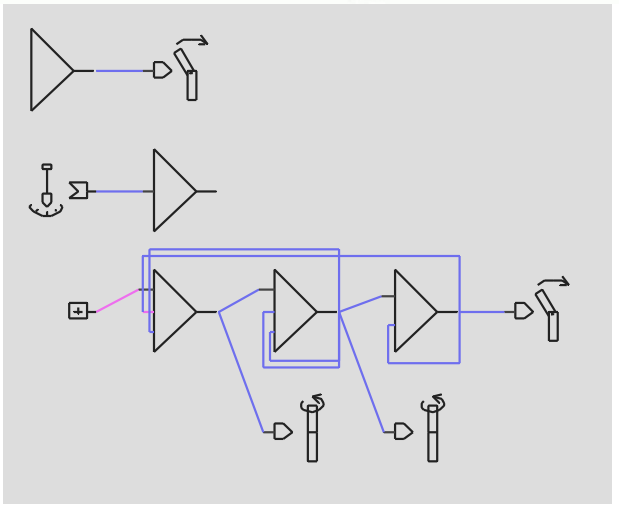






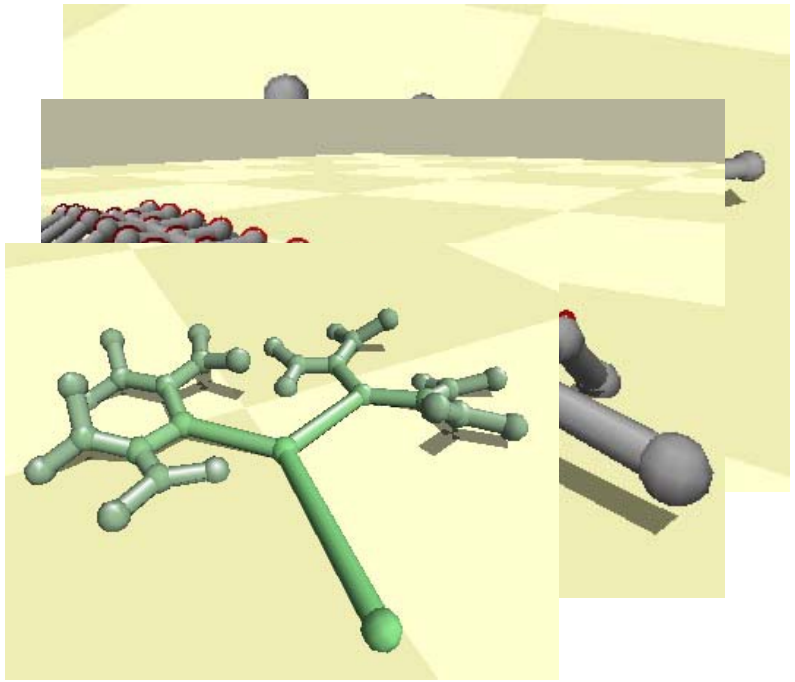


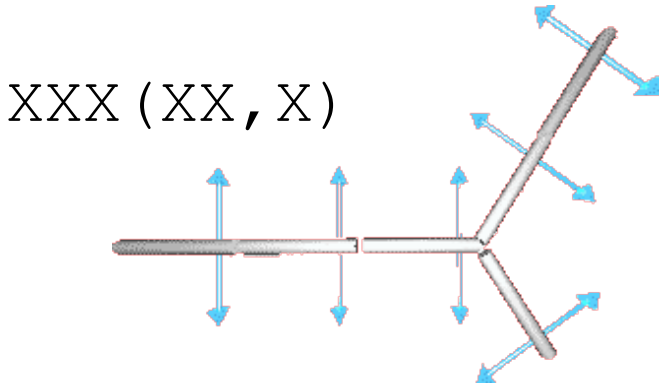


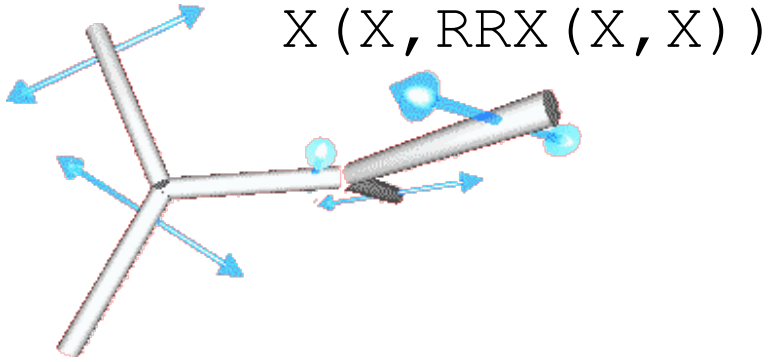


Recur representation

- properties are local, relative
- properties propagate along the body
- control elements (neurons, sensors) are near elements under control (muscles, sticks)
- recurrent body (tree)
- any topology of NN
- human-friendly





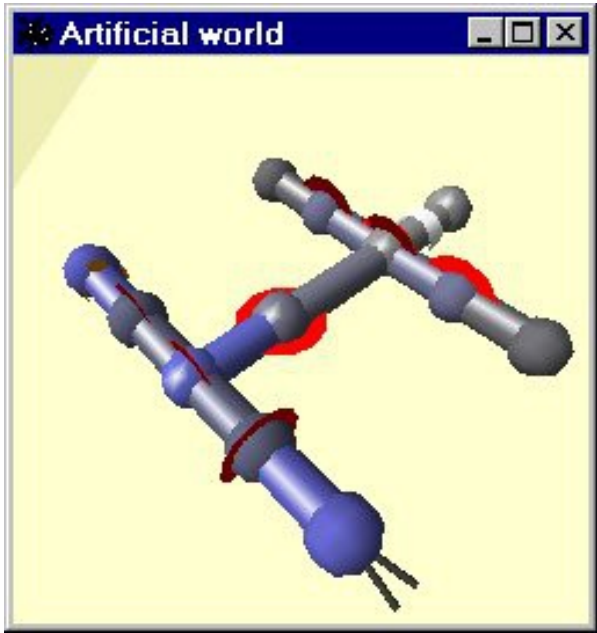


Recur ”modifiers”

`Rr`	Rotation of branching plane by ±45°
`Qq`	Skew of branching plane
`Cc`	Curvedness
`Ll`	Length
`Ff`	Friction
`Mm`	Muscle strength

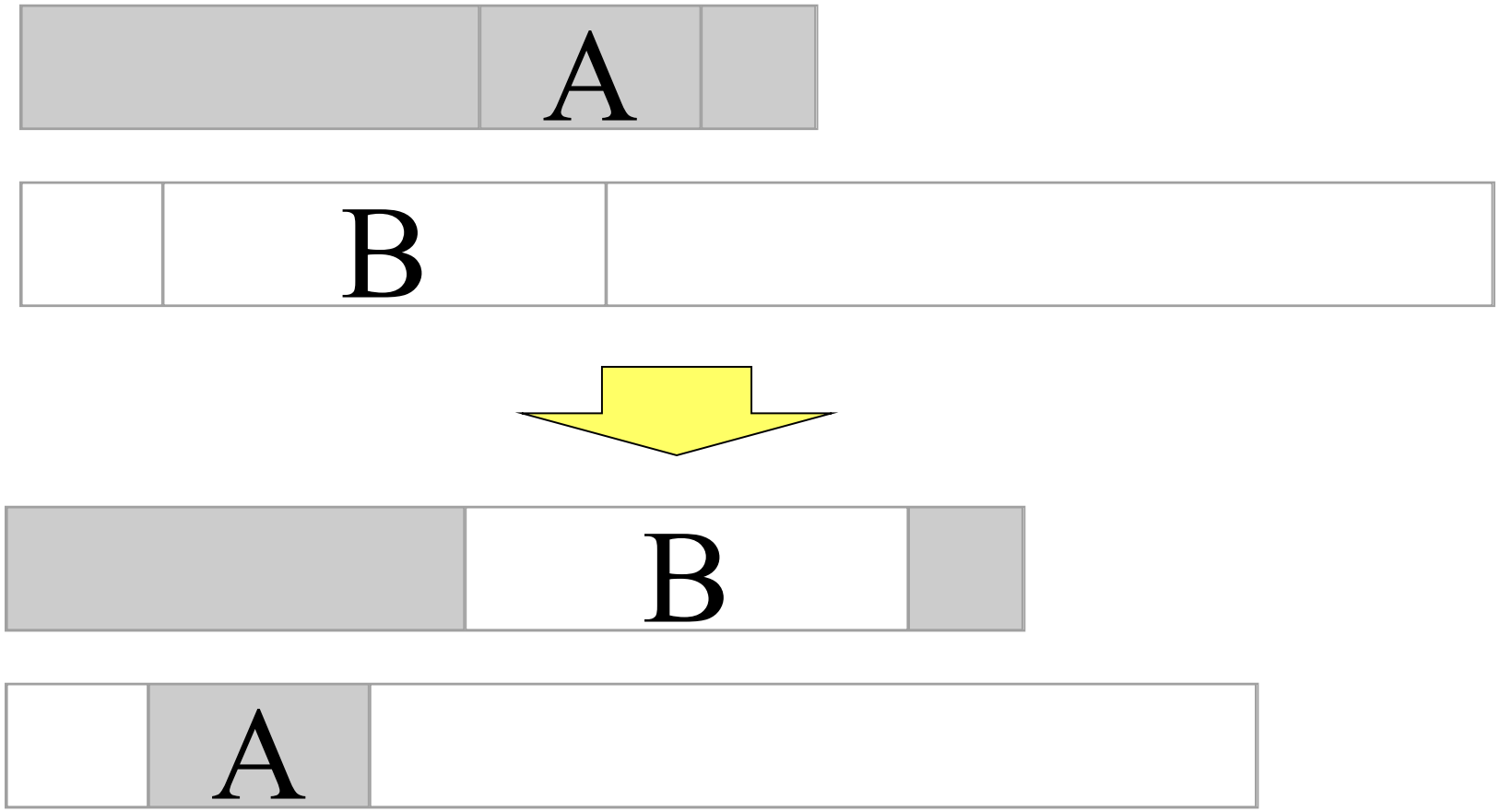
Recur example

```
lllfffX[0:2.420,2:-2,1 :-1][-1 :1,0:1,G:-1]
[-1:1](RRlllfffMMMX[|-1:-10]lllFFFMMMX
[|-2:-1],fffIXlllfffMMMsX[|6:10,*:-10]
(RRlllfffMMMIX[|-4:-10]lllFFFMMMIX[|-5:-1]
[S:313.590],,RRlllfffMMMIX[|-7:10]lllFFFMMMIX
[@-8:-1][T:1]),RRlllfffMMMX[|-10:10]
lllFFFMMMX[|-11:-1.784])
```



Show

Recur crossing-over



Gener representation

- Generative Lindenmayer system
- A set of production rules with parameters
- Rules are activated and generate a genome
- E.g.

```
10                      ← iterate 10 times
n0=56.000000            ← initial values for parameters
n1=55.000000
---                     ← production rules begin
P3
P3(n0,n1): n0>10.0 | X(2.000000)C(1.000000)R(2.000000)X(1.000000)?:n0>1 |
P3(12.000000-n1,2.000000)P2(3.000000,2.000000)
P2(n0,n1): n0>2.0 | X(3.000000)q(2.000000)X(1.000000)^P0(n0-
5.000000,n1)X(2.000000)X(1.000000)?
```

Simil representation

- BODY:
  - list of body components (sticks) with “links” and properties
  - joined according to links’ similarity
- BRAIN:
  - list of NN connections, effectors, senses with “links” and properties
  - connected according to links’ similarity

```
stick   ( 5, 9, 9, 6 ) ( 4, 7, 6, 5 ) 5 4 1 9 8 1 9 8 4 0 6
stick   ( 5, 9, 9, 8 ) ( 0, 7, 6, 6 ) 9 7 1 3 4 1 6 6 4 7 0
stick   ( 8, 0, 0, 2 ) ( 2, 4, 9, 3 ) 9 7 5 4 3 4 1 1 5 8 0
stick   ( 3, 9, 7, 2 ) ( 3, 2, 5, 5 ) 3 1 2 6 9 7 4 9 8 3 0

conn    ( 1, 1, 0, 8 ) | ( 9, 4, 0, 4 ) 5
conn T  ( 5, 1, 0, 3 ) @ ( 7, 8, 7, 7 ) 6
conn    ( 1, 1, 0, 8 ) | ( 9, 4, 6, 4 ) 5
```

Biol representation

- 26 characters of latin alphabet
- every sequence starting after **aa** and extending to the first **zz** sequence is considered a gene
- this encoding exhibits properties similar to DNA
- sample genome:
  `aaasdfgvcxaadzsdfgbvcxsfdrfgthnbzzvcxsdfgzz`
- Operators: horizontal gene transfer, crossing over, substitution, deletion, insertion, gene duplication, translocation

Chem representation

- “chemical” substances in 3D
- transforms initial substances into an organism
- rules of growth of body and brain
- saturation threshold to fire a rule
- propagation and changes of substances along growth directions
- propagation of properties of grown elements

4 rules
3 substances
2 properties

0.144 0.833 0.940, 0.546 0.249 grow stick, 0.859 0.604 0.707, 0.516 0.600
0.941 0.876 0.303, 0.038 0.630 grow stick, 0.902 0.320 0.035, 0.648 0.525
0.767 0.201 0.636, 0.751 0.022 grow stick, 0.321 0.661 0.663, 0.311 0.319
0.951 0.283 0.454, 0.428 0.997 grow stick, 0.996 0.554 0.162, 0.192 0.160

Messy representation

- any string of characters is a valid genotype
- simple genetic operators
- various interpretation approaches are possible, e.g.:

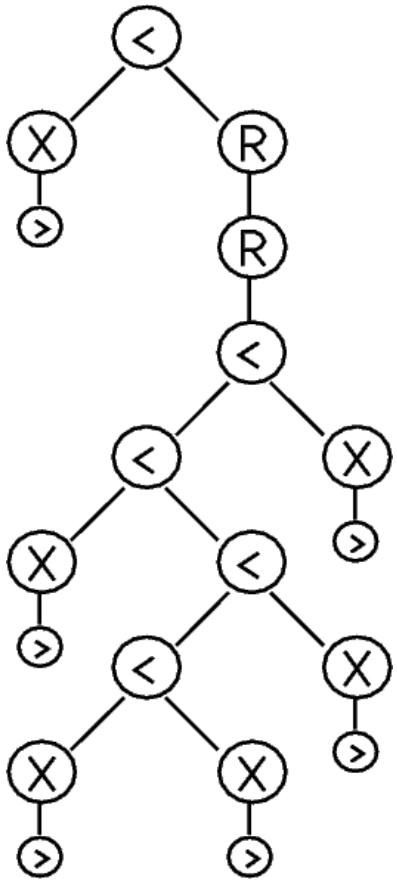
- sections correspond to elements of body and brain
- Z AAAAA BCLQU BCLQU BCLQU YYYYB BCNDG BCLQU BCLQU …
- SectionTag Z starts parts section:
- AAAAA – label, BCLQU BCLQU BCLQU … – coordinates
- Labels are recalculated as follows:
- AAAAA = 0*26^4 + 0*26^3 + 0*26^2 + 0*26^1 + 0*26^0 = 0,
- YYYYB = 24*26^4 + 24*26^3 + 24*26^2 + 24*26^1 + 1*26^0 = 11406097, …

Devel representation

- encodes development
- codes are commands of differentiation
- these instructions are executed in parallel
- supports symmetry and modularity
- development starts with a single, undifferentiated ancestor cell
- stops when all the cells are differentiated

<<Fm<<X>N[1:-2.25791]|[G:3.49452]>X>M,X>X

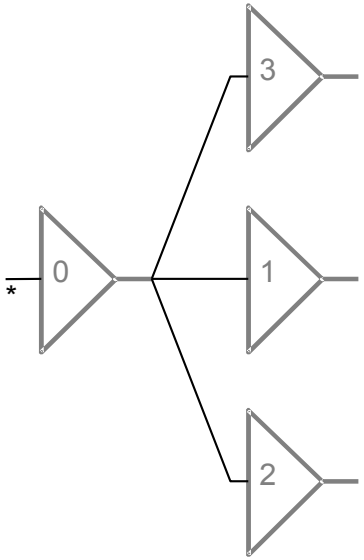
Devel development



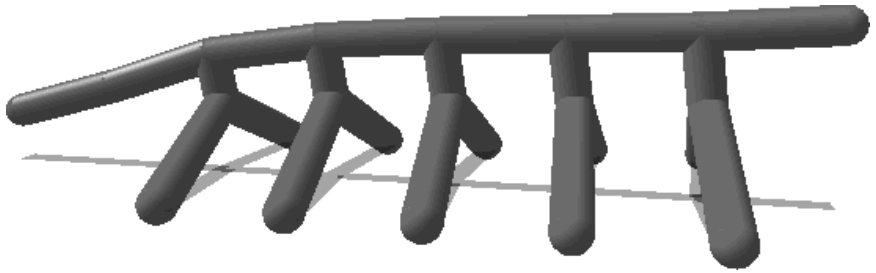
<X>RR<<X><<X>X>X>X

Devel examples

- Neural links are duplicated when a neuron divides



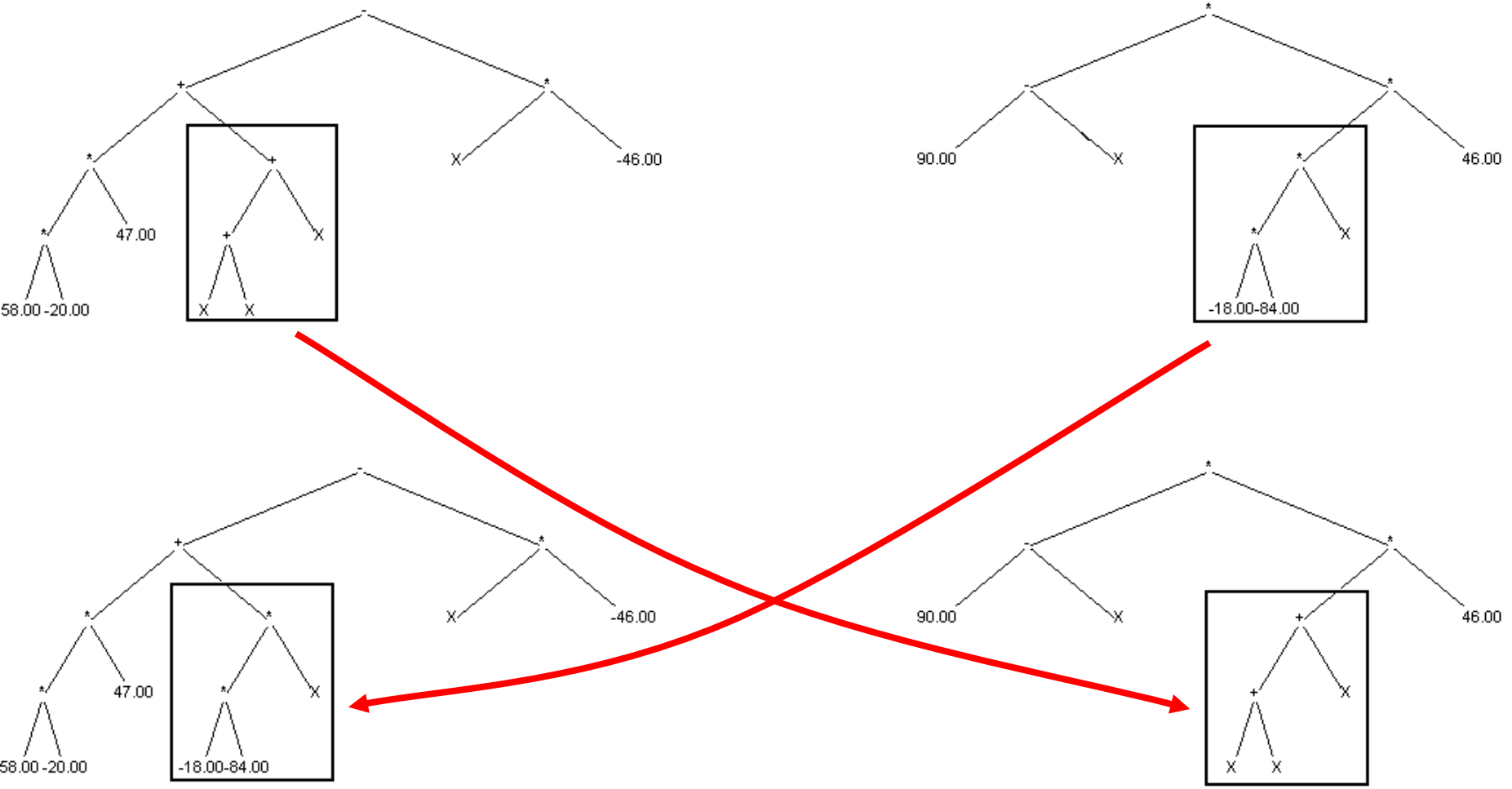
- Parts of the genome may be repeated



rr<X>#5<,<X>RR<<ll
X>LX>LX>>X

<X>N<[*:5]>[-1:5]<><>>

Devel crossing-over



Mutation and repair

- Mutation: modification of every element of a genotype. Small, local changes
- Validity test: many aspects
- Repair: attempt to correct an invalid genotype
  - ensure each property value is within allowed interval
  - correct neural links
  - contextual: match brackets, etc.
  - ...