Cancer classification based on miRNA profiles using ASP

K. Becker and H. Klarner

Freie Universität Berlin, Germany

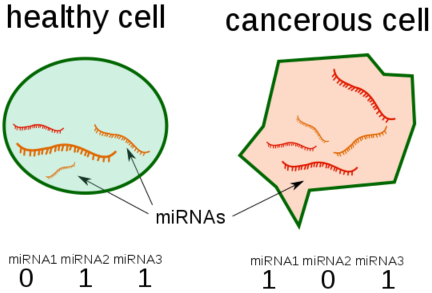
Berlin, Mai 2016

Freie Universität Berlin

Bundesministerium für Bildung und Forschung

Selective cell targeting

- **Problem:** Discrimination of tumor from healthy tissues



- **Idea:** Cells differ in miRNA profiles

In vitro classification - Biochemical circuits

- synthesized in laboratory

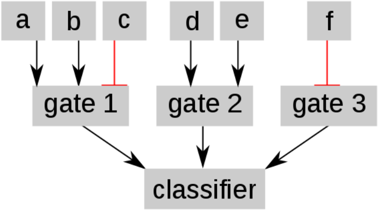
A boolean expression in conjunctive normal form (CNF)

- conjunction (AND) of *gates*
- each gate is a disjunction (OR) of literals
- **example:**

$$(a + b + !c) * (d + e) * (!f)$$

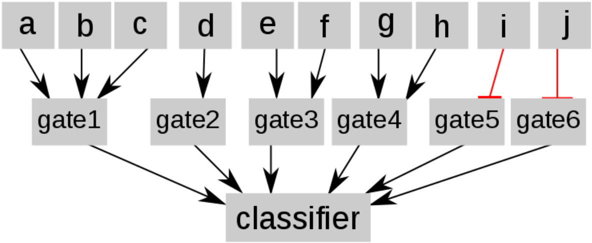
$+$ means disjunction, $*$ means conjunction, ! means negation

- CNF evaluates to 1 (*predicts cancer*) iff every gate evaluates to 1



Constraints from biology

- less than 10 inputs in total
- no more than 6 inputs attached to the AND gate
- no more than 3 inputs atttached to any OR gate
- no NOT gates attached to an OR gate
- no more than 2 OR gates
- no more than 4 NOT gates



Input: Data

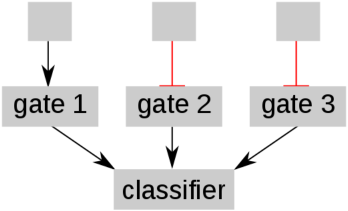
		cancer?	miRNAs		
	ID	Annots	g1	g2	g3
tissues	1	0	1	1	0
	2	0	0	0	1
	3	1	0	1	0

```
tissue(1,healthy). tissue(2,healthy). tissue(3,cancer).

data(1,g1,high). data(1,g2,high). data(1,g3,low).
data(2,g1,low).  data(2,g2,low).  data(2,g3,high).
data(3,g1,low).  data(3,g2,high). data(3,g3,low).

is_mirna(Y) :- data(X,Y,Z).
```

Input: Classifier structure



```
is_gate_type(type1).

upper_bound_pos_inputs(type1, 1).
upper_bound_neg_inputs(type1, 0).
lower_bound_pos_inputs(type1, 0).
lower_bound_neg_inputs(type1, 0).
upper_bound_gate_occurence(type1, 1).

upper_bound_inputs(2).
```

Decision 1: Number of gates

```
1 {number_of_gates(1..2)} 1.

is_integer(1..2).
is_gate_id(GateID) :- number_of_gates(X),is_integer(GateID),
                      GateID<=X.
```

Decision 2: Gate types

```
1 {gate_type(GateID, X) : is_gate_type(X)} 1 :- is_gate_id(GateID).
```

Decision 3: Inputs for each gate

positive inputs:

```
X{gate_input(GateID,positive,MiRNA):is_mirna(MiRNA)}Y
   :-is_gate_id(GateID),gate_type(GateID,GateType),
     lower_bound_pos_inputs(GateType,X),
     upper_bound_pos_inputs(GateType,Y).
```

negative inputs:

```
X{gate_input(GateID,negative,MiRNA):is_mirna(MiRNA)}Y
   :-is_gate_id(GateID),gate_type(GateID,GateType),
     lower_bound_neg_inputs(GateType,X),
     upper_bound_neg_inputs(GateType,Y).
```

Constraints

- conditional count constraints
-

Gates must have inputs

1 {gate_input(GateID, Sign, MiRNA):

```
1 {gate_input(GateID, Sign, MiRNA):
    is_sign(Sign), is_mirna(MiRNA)} :- is_gate_id(GateID).
```

Inputs must be unique

```
{gate_input(GateID,Sign,MiRNA):
  is_sign(Sign), is_gate(GateID)} 1 :- is_mirna(MiRNA).
```

Number of inputs is bounded

```
{gate_input(GateID, Sign, MiRNA):
  is_gate_id(GateID), is_sign(Sign), is_mirna(MiRNA)} X :-
upper_bound_inputs(X).
```

Occurences of gates

```
{gate_type(GateID,GateType):
   is_gate_type(GateType),
   is_gate_id(GateID)} X :-
  upper_bound_gate_occurence(GateType,X).
```

Activity of gates

```
gate_fires(GateID,TissueID) :-
  gate_input(GateID,positive,MiRNA),
  data(TissueID,MiRNA,high).

gate_fires(GateID,TissueID) :-
  gate_input(GateID,negative,MiRNA),
  data(TissueID,MiRNA,low).
```

The classifier

```
classifier(TissueID, healthy) :-
  not gate_fires(GateID, TissueID),
  is_gate_id(GateID), is_tissue_id(TissueID).

classifier(TissueID, cancer) :-
  not classifier(TissueID, healthy),
  is_tissue_id(TissueID).
```

Optimization

- single objective
- with priorities
- weighted sum

Single Objective

```
#minimize{ 1@1,GateID:gate_input(GateID,Sign,MiRNA) }.
#minimize{ 1@2,MiRNA: gate_input(GateID,Sign,MiRNA) }.
```

Priorities

```
#minimize{ 1@1,GateID:gate_input(GateID,Sign,MiRNA) }.
#minimize{ 1@2,MiRNA: gate_input(GateID,Sign,MiRNA) }.
```