Neural Grammar Networks
in
QSAR Chemistry

Eddie Y.T. Ma

Department of Biology
University of Waterloo
Waterloo, Ontario, Canada
e3ma@uwaterloo.ca

Stefan C. Kremer

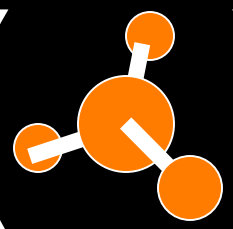
Department of Computing and Information Science
University of Guelph
Guelph, Ontario, Canada
skremer@uoguelph.ca

Overview

- Background
  - Quantitative Structure-Activity Relationship (QSAR)
  - Computing Background
- The Neural Grammar Network
- Performance

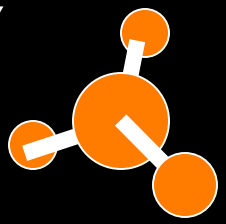
QSAR

- ✓ Predicting a quantity of biological action (range) for a suite of molecules (domain)
- ✓ Example biological actions: Toxicity, Mutagenicity, Binding affinity.

qsar() = 92.1

QSAR - Why?

- ✓ Inexpensive way to prescreen molecules before expensive biomedical assays
- ✓ Better QSAR methods are constantly sought to improve performance and reduce costs

qsar() = 92.1

QSAR models usually consist of two parts

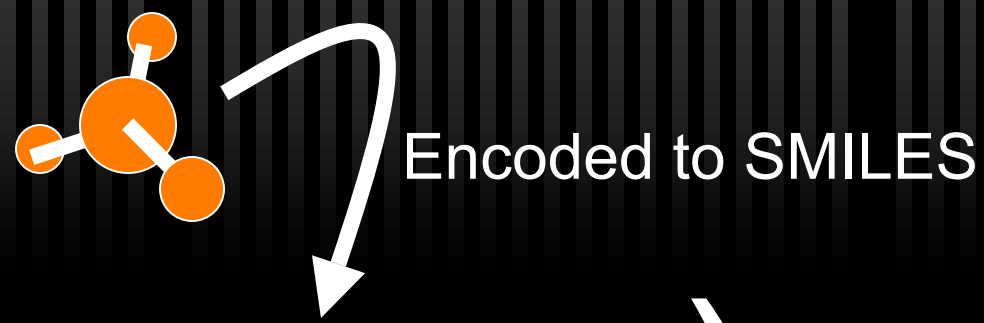
1. Input Descriptors (real-value vectors, expert)
   - ✓ Encodes physical properties, each molecule
2. Learning device (*Ld*)

Encode to a Descriptor (LOSSY!)

$$Ld(\langle 810, 3.2, 6, 9.93 \rangle) = 92.1$$

QSAR – our model, an experiment

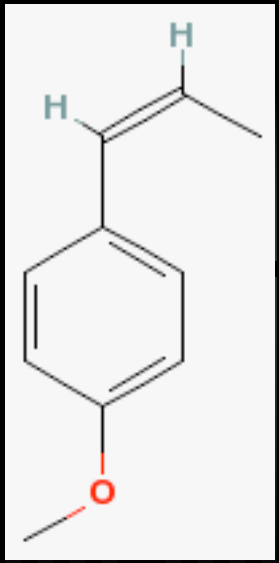
- ✓ Cheminformatics strings used instead of descriptors.



$$f_?(\text{“CC(C)C=CO”}) = 92.1$$

Example SMILES and InChI (1of2)

1-Methoxy-4-(1-propenyl)benzene



SMILES:

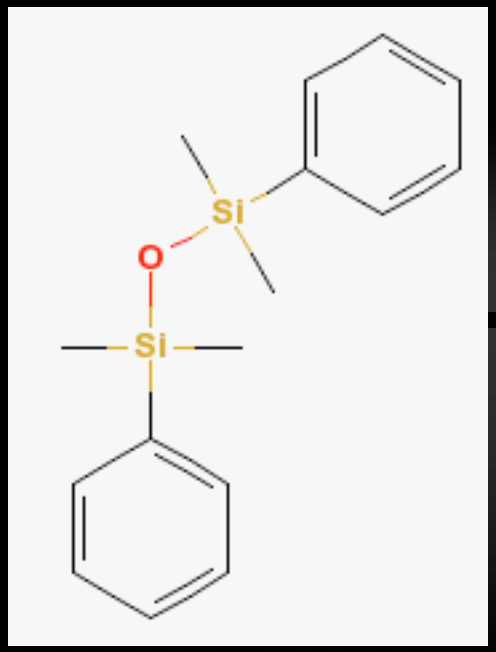
`C(=C/C)\c1ccc(cc1)OC`

InChI:

`InChI=1/C10H12O/c1-3-4-9-5-7-10(11-2)8-6-9/h3-8H,1-2H3/b4-3+`

Example SMILES and InChI (2of2)

1,3-Diphenyltetramethyldisiloxane



SMILES:

O([Si@@](c1ccccc1)(C)C)[Si@@](c1ccccc1)(C)C

InChI:

InChI=1/C16H22OSi2/c1-18(2,15-11-7-5-8-12-15)17-19(3,4)16-13-9-6-10-14-16/h5-14H,1-4H3

What form should our model take...

- ✓ A device that can mine the formal syntax of SMILES and InChI as input

Concrete formal language example

"1+3•2÷0"
"1-2-3+5÷5"
"3"
"3•3"
"8÷1"
"0•0+0"

- ✓ Syntax features
  - ✓ Tokens
  - ✓ Structure
  - ✓ Operator precedence
  - ✓ Nested statements

The matching formal grammar

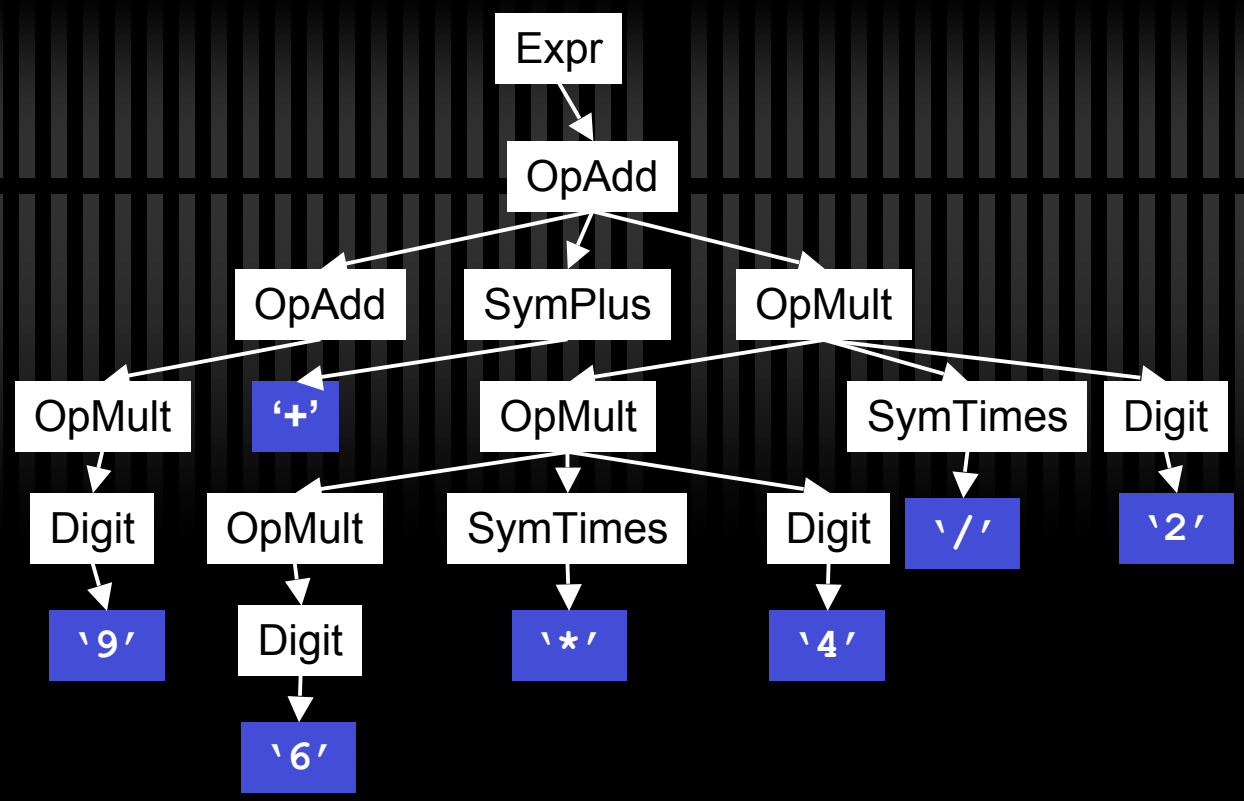
```
Expr -> OpAdd
OpAdd -> OpAdd SymPlus OpMult | OpMult
SymPlus -> `+' | '-'
OpMult -> OpMult SymTimes Digit | Digit
SymTimes -> `*' | '/'
Digit -> `0' ... `9'
```

A parse tree is statement structure

Example Statement:

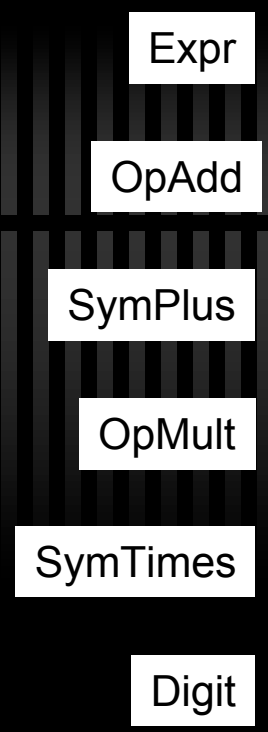
"9+6•4/2"

Parse Tree:

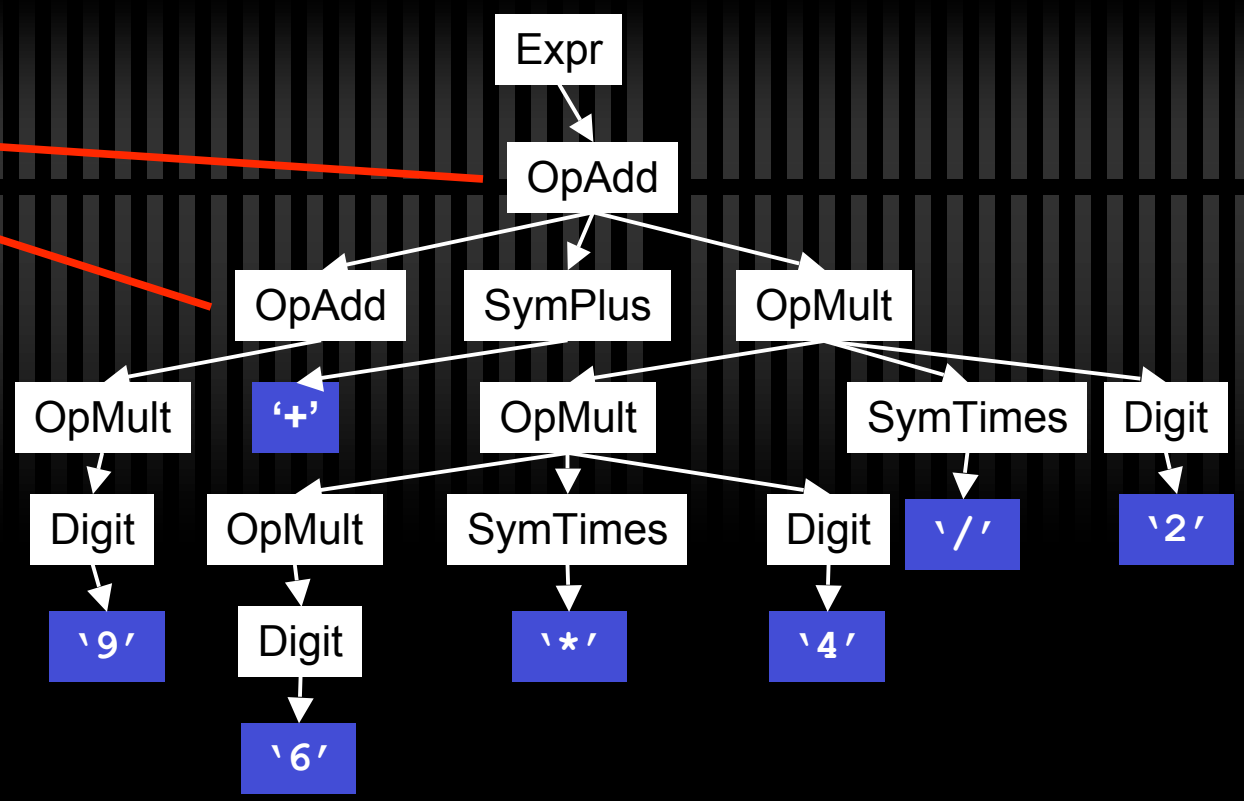


Pieces of the grammar correspond to functional parts.

Reusable Functional Parts:



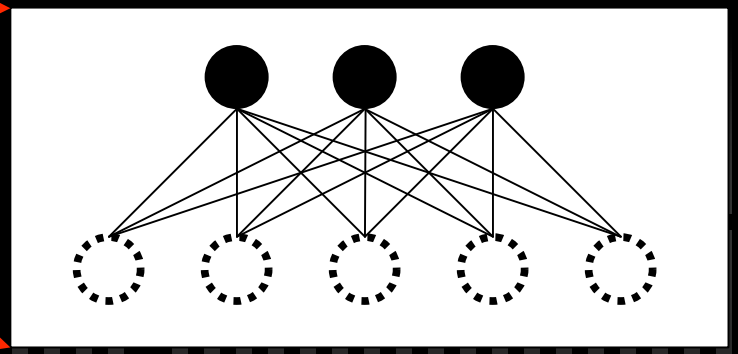
Parse Tree:



Our model maps neural layers to the parse tree.

Reusable Functional Parts:

- Expr
- OpAdd
- SymPlus
- OpMult
- SymTimes
- Digit



What if each functional part were mapped to a neural network layer?

Why Neural Networks?

Artificial Neural Networks

- ✓ Universal function approximators
- ✓ Input (domain) and output (range) must be expressed as real-value vectors

ann(< 0.8, 0.2, 0.55, 0.72 >) = < 0.24, 0.67 >

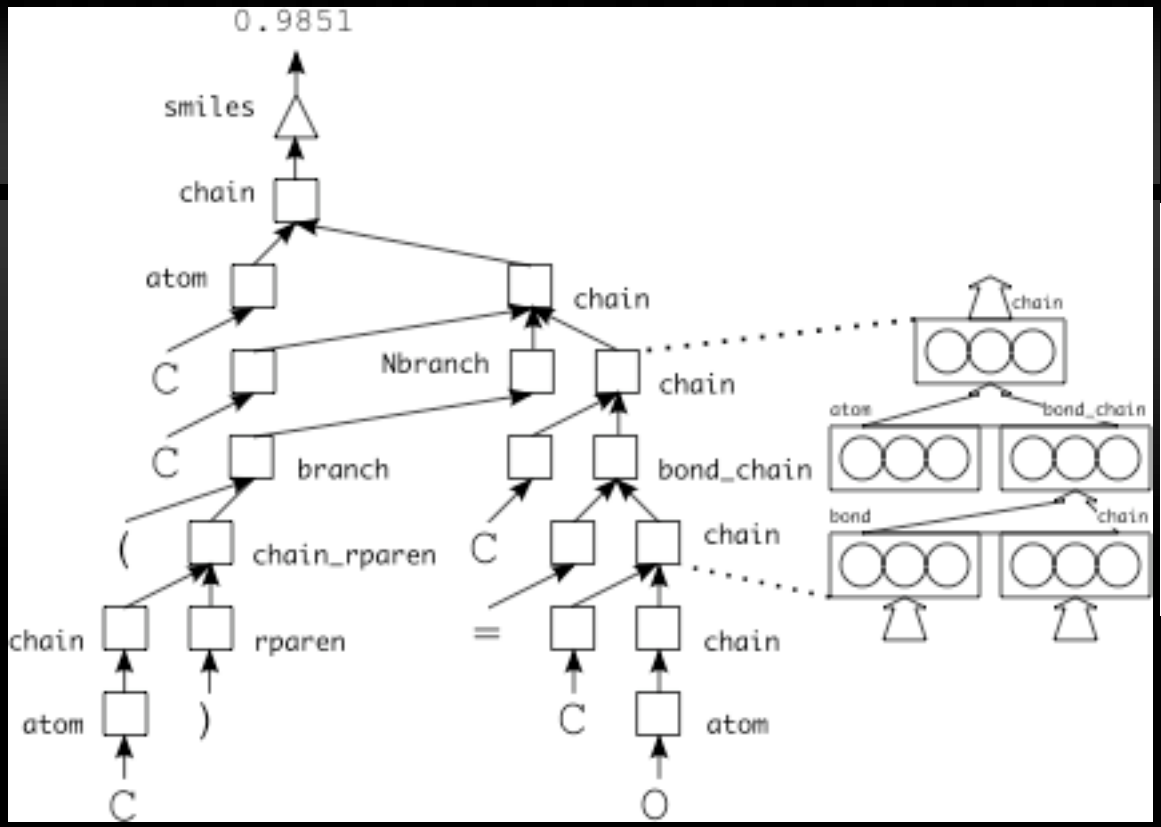
And Finally…

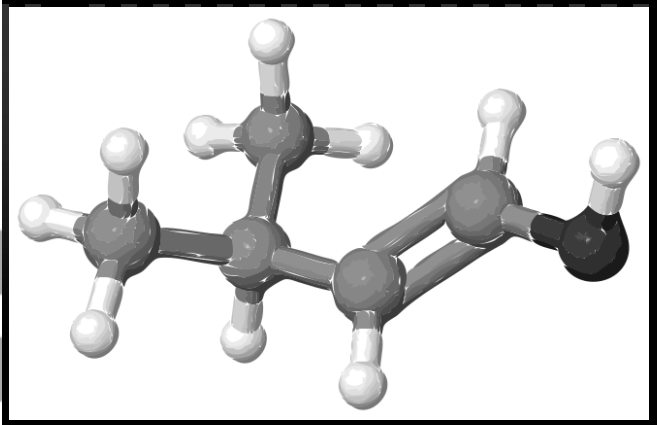
The NEURAL GRAMMAR NETWORK

- ✓ Use formal string structure as topology of neural network
- ✓ Assemble a custom NGN for each example string by snapping together the reusable components
- ✓ Use neural networks' learning algorithm and general function approximation ability

A SMILES-Neural Grammar Network

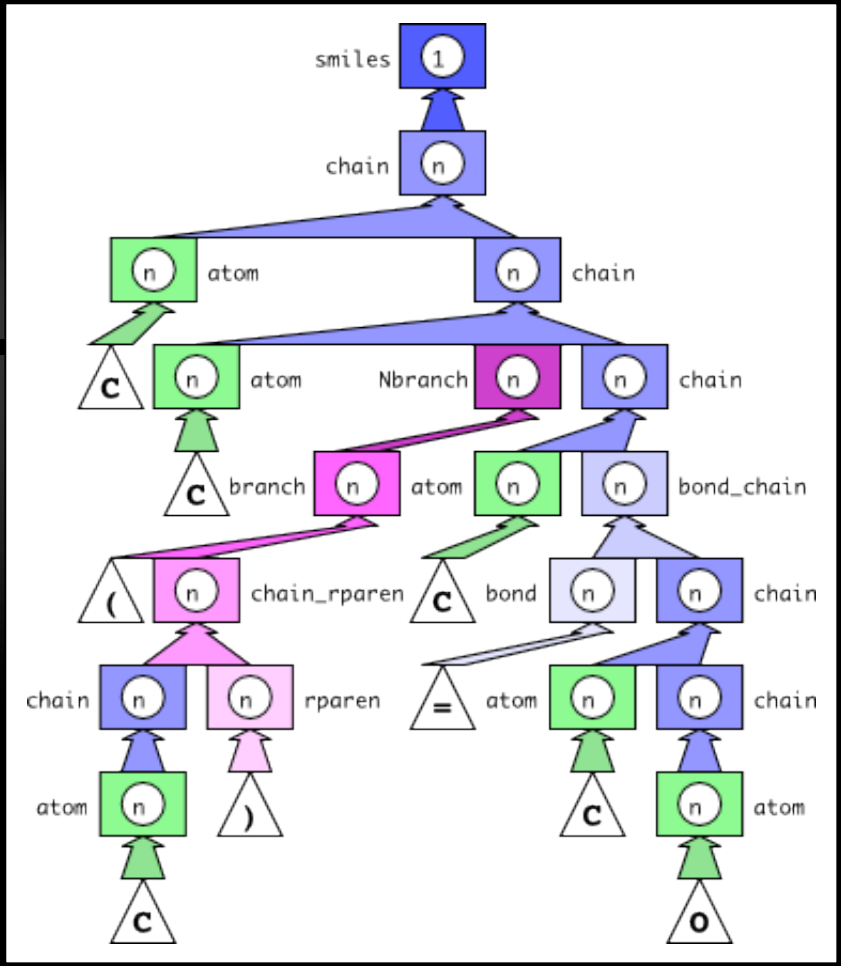
- ✓ SMILES isopentenol example "CC(C)C=CO"



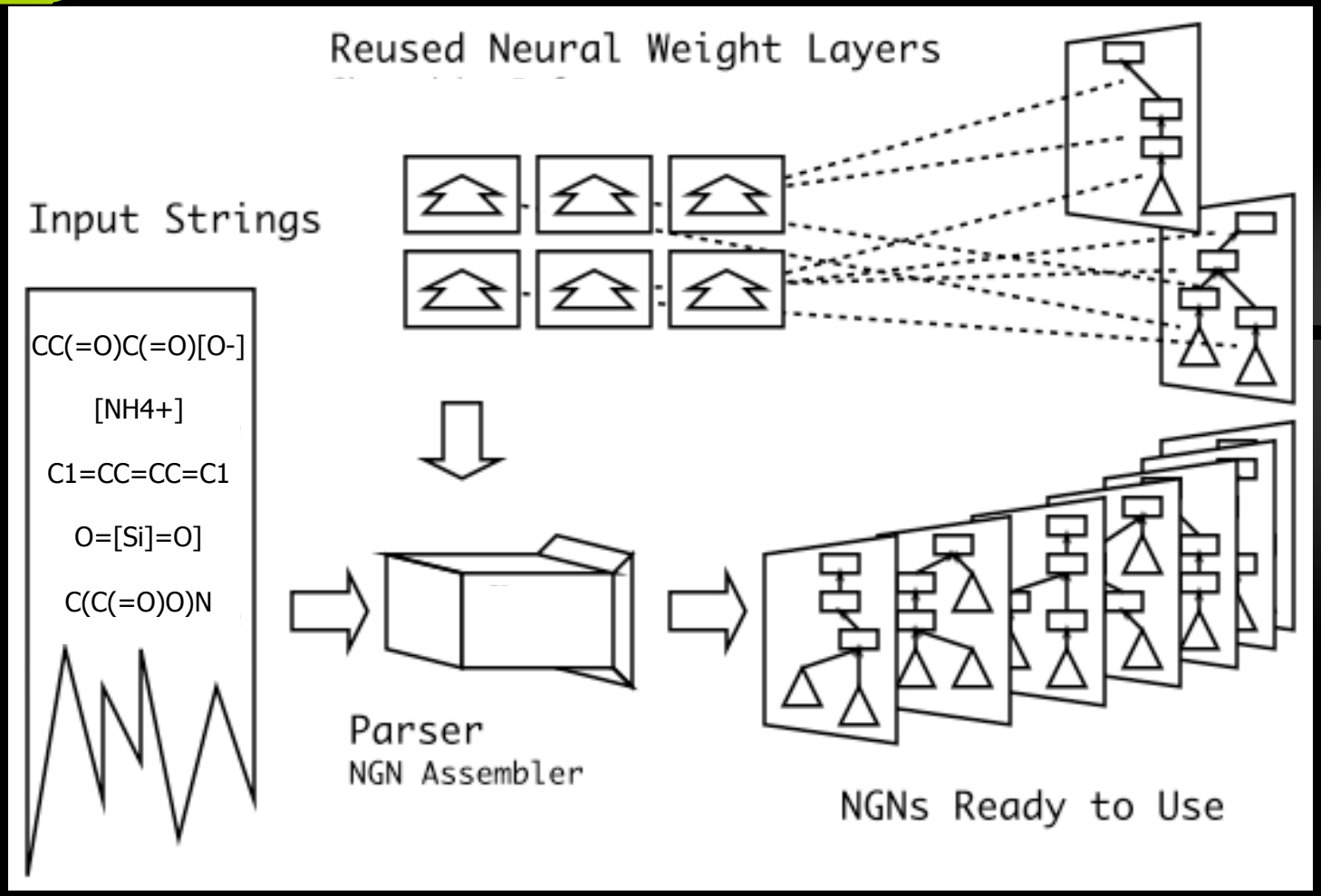


Structured Shared Network Components

CC(C)C=CO



Emphasis: Parts reused between strings-- Training, Prediction functions accomplished!



Experiments and Results

- ✓ Experimental design
- ✓ Results for QSAR studies

PERFORMANCE: classification and regression

- Classification (a.k.a. “Matching to a Category”)
  - Learn and predict what objects fall into what category
  - In QSAR, usually binary
- Regression (a.k.a. “Fitting onto a Curve”)
  - Learn and predict the mapping of objects onto a scalar range
  - In QSAR, usually log-normalized scale

PERFORMANCE: classification and regression

- Splitting datasets
  - Leave-20%-out cross validation design
  - Replicate previous designed test sets

Internal and External Validity!

PERFORMANCE: classification datasets

Dataset	Dataset Full Name	Size	N^+	N^-	Reference
BZR	Benzodiazepine Receptor	405	230	175	Sutherland et al. (2003)
Cox2	Cyclooxygenase 2	467	273	194	Sutherland et al. (2003) Kauffman and Jurs (2001)
DHFR	Dihydrofolate Reductase	756	302	454	Sutherland et al. (2003)

Best work was researched and used as a point of comparison

PERFORMANCE: classification evaluation

Accuracy $$Q = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Sensitivity $$\mathrm{SE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Specificity $$\mathrm{SP} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

PERFORMANCE: regression datasets

Dataset	Dataset Full Name	Size	Reference
ACE	Angiotensin Converting Enzyme	114	Sutherland et al. (2004) Depriest et al. (1993)
Cox2	Cyclooxygenase-2	282	Sutherland et al. (2004)
Therm	Thermolysin	76	Sutherland et al. (2004) Depriest et al. (1993)
Thr	Thrombin	88	Sutherland et al. (2004) Bohm et al. (1999)
AChE	Acetylcholinesterase	111	Sutherland et al. (2004)
GPB	Glycogen Phosphorylase B	66	Sutherland et al. (2004) Gohlke and Klebe (2002)
BZR	Benzodiazepine Receptor	163	Sutherland et al. (2004)
DHFR	Dihydrofolate Reductase	397	Sutherland et al. (2004)

A range of small to medium datasets, n = [66, 397].

PERFORMANCE: regression evaluation

$$\mathrm{SD} = \sum_{i=1}^{N_{\mathrm{test}}} (y_i - \bar{y}_{\mathrm{test}})^2$$

$$\mathrm{PRESS} = \sum_{i=1}^{N_{test}} (y_i - \mathrm{prediction}(x_i))^2$$

$$r^2_{\mathrm{pred}} = \frac{\mathrm{SD - PRESS}}{\mathrm{SD}}$$

PERFORMANCE: classification results

- State of the art performance on 3 experiments
  - DHFR designed test set
    - $Q_{InChI\text{-}NGN}$ = 73.2% vs. Q_{SIMCA} = 75.5%
  - DHFR cross validation
    - $Q_{InChI\text{-}NGN}$ = 74.8% vs. Q_{SFGA} = 64.5%
  - BZR cross validation
    - $Q_{InChI\text{-}NGN}$ = 69.9% vs. Q_{SIMCA} = 71.5%

SIMCA, SFGA are models described by Sutherland et al.

PERFORMANCE: classification results

DHFR

Design	Method	Q(%)	SE(%)	SP(%)
Leave-20%-Out	SIMCA	63.5±9.5	57±10	70±9
	RP	61±12	57±12	65±12
	SFGA	64.5±10.5	65±11.0	64±10.0
	InChI-NGN	74.8±1.63	70.3±2.44	77.5±1.72
40% Test Set	SIMCA	75.5	74	71
	RP	65	57	73
	SFGA	68.5	71	66
	InChI-NGN	73.2	73.1	100.0

BZR

Design	Method	Q(%)	SE(%)	SP(%)
Leave-20%-Out	SIMCA	71.5±11.0	73±10	70±12
	RP	65.5±12	68±12	65±12
	SFGA	68.5±12	69±11	68±13
	InChI-NGN	69.9±1.98	73.4±1.87	65.3±2.29
40% Test Set	SIMCA	72	68	76
	RP	69	64	74
	SFGA	75.5	70	81
	InChI-NGN	63.2	62.1	64.6

Cox2

Design	Method	Q(%)	SE(%)	SP(%)
Leave-20%-Out	SIMCA	78±9	79±9	77±9
	RP	69.5±12	72±12	67±12
	SFGA	74±9.5	76±9	72±10
	InChI-NGN	72.2±1.36	74.4±1.01	68.7±2.45
40% Test Set	SIMCA	71	75	67
	RP	71	79	63
	SFGA	73.5	75	72
	InChI-NGN	65.1	62.5	68.4

PERFORMANCE: regression results

- *Outperforms* on six datasets
  - GPB, ACE, AChE, Cox2, Thr and DHFR
  - Performance variance is high however

Data set	SMILES-NGN	InChI-NGN	CoMFA	HQSAR	MK
GPB	0.79±0.23	0.48±2.90	0.42	0.58	—
ACE	0.74±0.46	0.78±0.31	0.49	0.30	0.58
AChE	0.68±0.80	0.60±0.78	0.47	0.37	0.50
Cox2	0.56±1.33	0.37±4.28	0.29	0.27	—
Therm	0.47±2.72	0.52±1.59	0.54	0.53	—
BZR	-0.29±17.30	0.11±8.74	0.00	0.17	0.36†
Thr	—	0.70±0.72	0.63	-0.25	—
DHFR	—	0.66±0.94	0.59	0.63	0.65†

PERFORMANCE: regression results

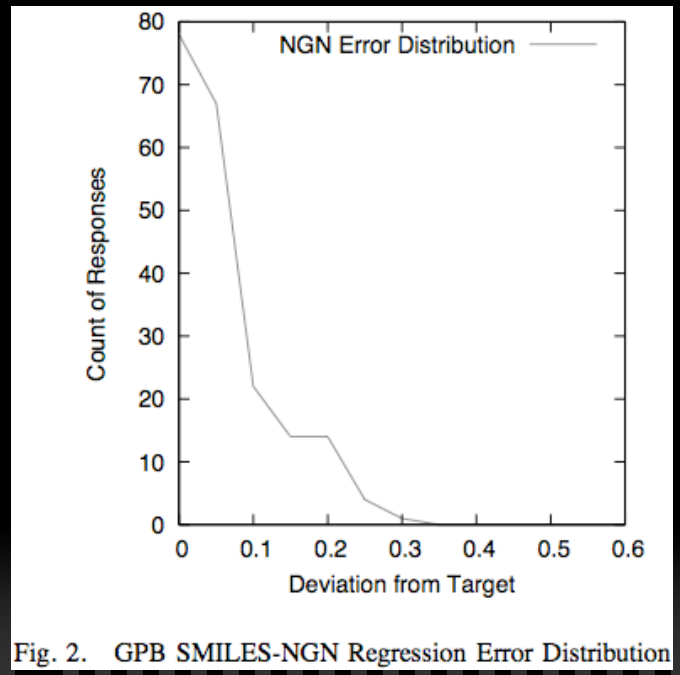


Fig. 2. GPB SMILES-NGN Regression Error Distribution

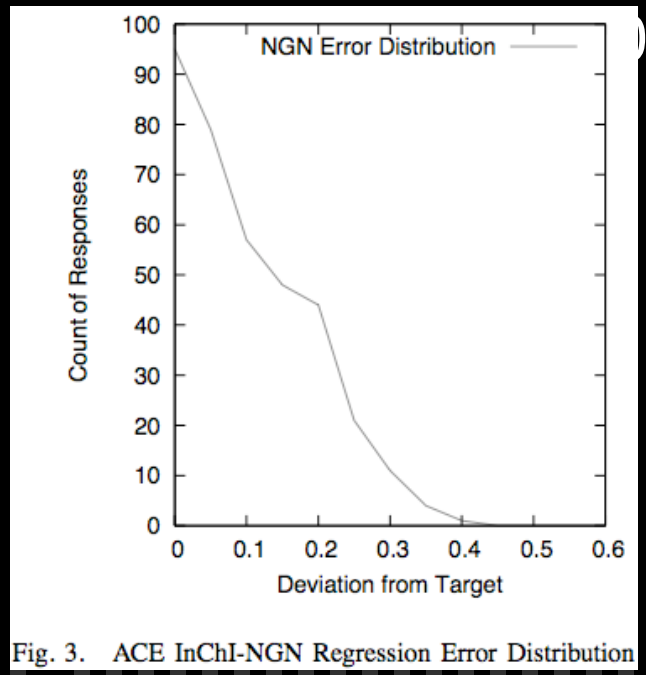


Fig. 3. ACE InChI-NGN Regression Error Distribution

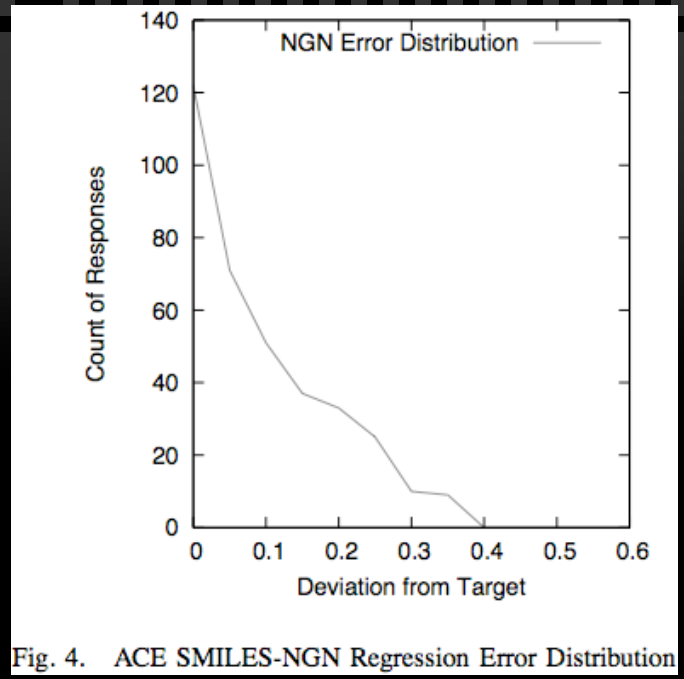


Fig. 4. ACE SMILES-NGN Regression Error Distribution

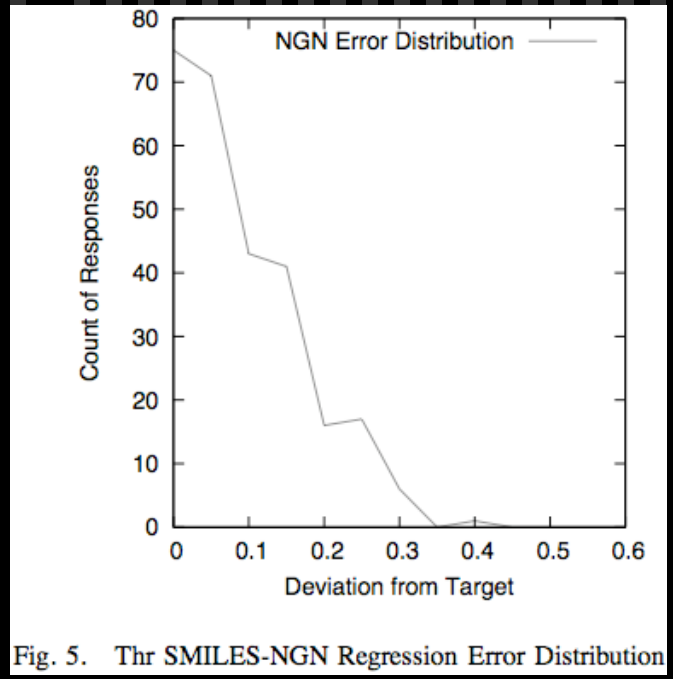


Fig. 5. Thr SMILES-NGN Regression Error Distribution

Benefits

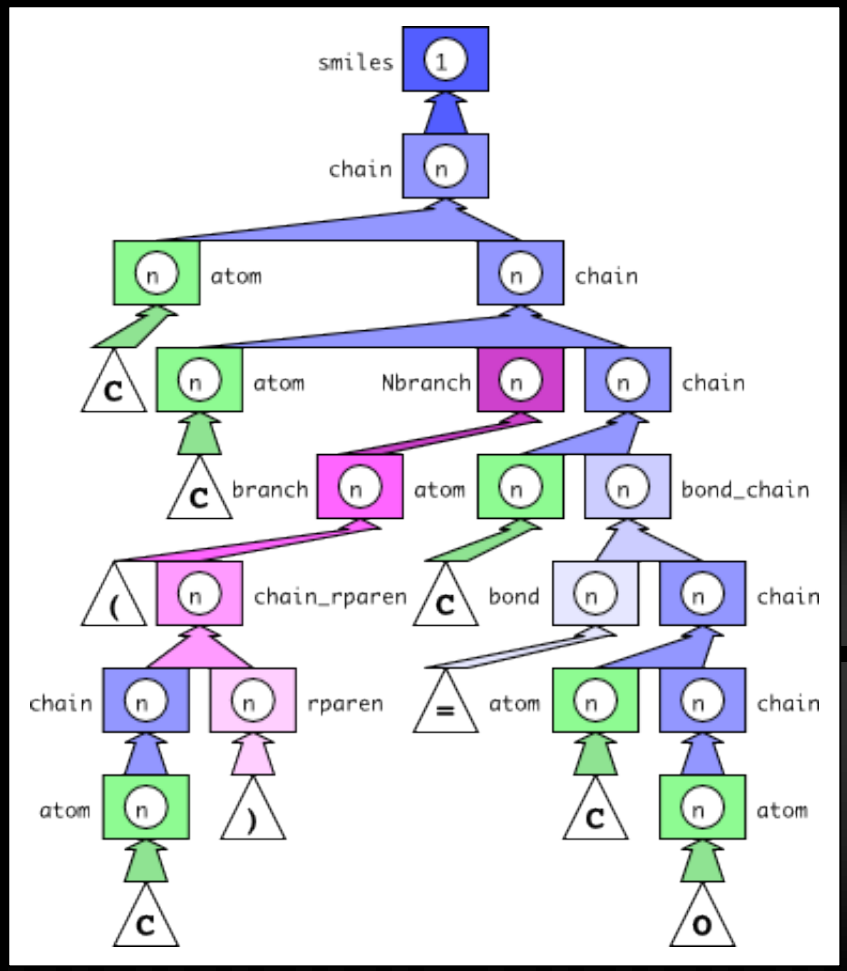
- ✓ No expert knowledge is needed in lossy descriptor selection
- ✓ Fully represents a traversal of a molecule
- ✓ Leverages developed freely accessible languages (SMILES, InChI)

CONCLUSION and future

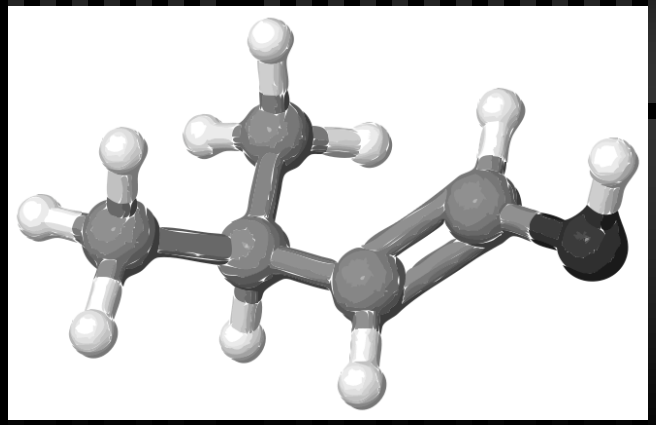
- ✓ The NGN has been presented for formal string classification and regression
- ✓ An NGN system has been applied to QSAR
- ✓ State of the art classification performance
- ✓ Superior regression performance (although standard deviation is high)
- ✓ This from a prototype NGN system!

CONCLUSION and future

- Should be tried in more QSAR problems
- New problems with formal string domain and new grammars
  - e.g. Image Processing
- Other advanced training and recurrent data treatment possible!



Thanks Everyone!



CC(C)C=CO

InChI=1S/C5H10O/c1-5(2)3-4-6/h3-6H,1-2H3/b4-3+