Tutorial, ICANN 2007, Porto

Evolving Connectionist Systems: The Knowledge Engineering Approach

Nikola Kasabov
Knowledge Engineering and Discovery Research Institute, KEDRI
Auckland University of Technology (AUT)
Auckland, New Zealand

nkasabov@aut.ac.nz, www.kedri.info

Abstract

- Evolving Connectionist Systems (ECOS) are neural networks that **develop their structure, functionality and internal knowledge representation** through continuous learning from data and interaction with the environment. ECOS can also evolve through generations of populations using evolutionary computation, but the focus of the tutorial is on the adaptive learning and improvement of each individual system. The learning process can be: on-line, off-line, incremental, supervised, unsupervised, active, sleep/dream, etc. These general principles can be applied to develop different models of computational intelligence - evolving simple connectionist systems, evolving spiking neural networks, evolving rule based and fuzzy systems, evolving kernel-based systems, evolving quantum-inspired systems, and many different integrated, hybrid models [1].
- The emphasis here is on the knowledge engineering aspect of the systems, i.e. how to represent human knowledge in a system and to extract interpretable information that can be turned into knowledge.
- ECOS are demonstrated on several challenging problems from bioinformatics, neuroinformatics, neuro-genetics [2], medical decision support, autonomous robot control, adaptive multimodal information processing.
- The tutorial targets computer scientists, neuroscientists, biologists, engineers, both researchers and graduate students an follows the structure of the book [1].

- [1] N.Kasabov, Evolving connectionist systems: The Knowledge Engineering Approach, Springer, 2007
- [2] L.Benuskova and N.Kasabov, Computational Neurogenetic Modelling, Springer, 2007

- **Keywords:** Computational Intelligence, Neuroinformatics, Bioinformatics, Knowledge-based neural networks, Evolving connectionist systems, Data Mining; Knowledge Discovery

Content

Part I: Methods

Part II. Applications

1. **Evolving Intelligent Systems: Introduction**

Evolving process: the process is *unfolding, developing, revealing, changing over time* in a continuous way

EIS: An information system that develops its structure and functionality in a continuous, self-organised, adaptive, interactive way from incoming information, possibly from many sources, and performs intelligent tasks (e.g. adaptive pattern recognition, decision making, concept formation, languages,….).

The principle of **evolvability.**

EIS is characterised by:

- Adaptation in an incremental mode (possibly, on-line, life-long)
- Fast learning from large amount of data, e.g. possibly 'one-pass' training
- Open structure, extendable, adjustable
- Memory-based (add and retrieve information, delete information, trace the system development)
- Active interaction with other systems and with the environment
- Represent adequately space and time at their different scales
- Knowledge-based: rules
- Self-improvement

What are the rules that make a process evolving?
Can a system, that evolves from data measuring this process, capture these rules, for the time?
How do we turn these rules into discoveries?

Inspiration from Biology

- Multilevel Information processing
- Complex interaction between different levels of information processing
- Everything is evolving
- How do we model it?
- Dynamic Integration of Knowledge and Modeling techniques
- The principle of evolvability

6. Evolutionary (population/generation) processes

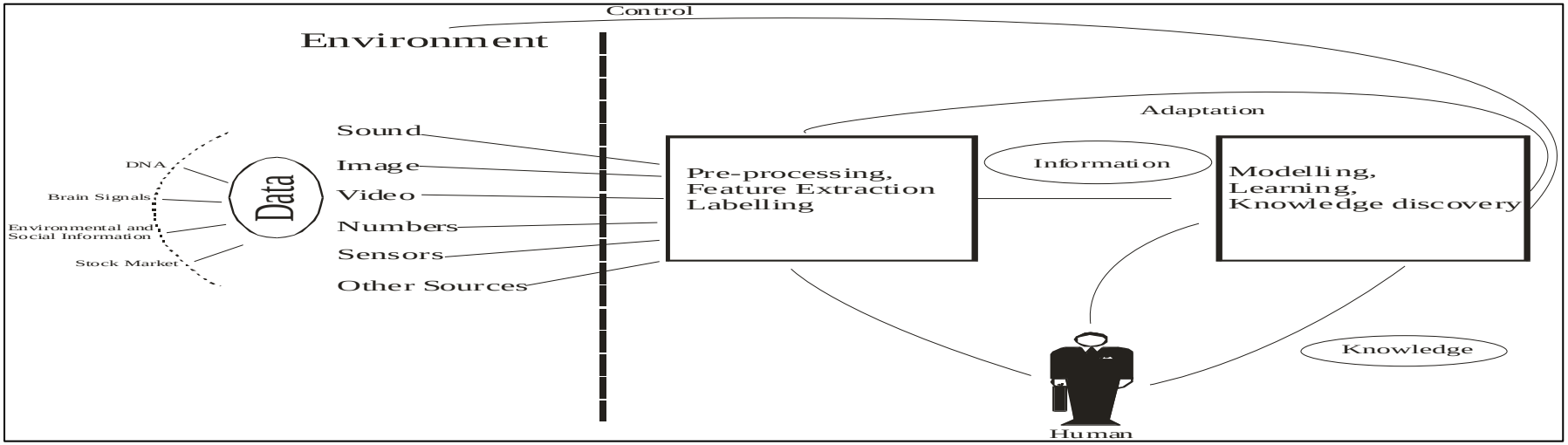
5. Brain cognitive processes

4. System information processing (e.g. neuronal ensembles)

3. Information processing in a cell (neuron)

2. Molecular information processing (genes)

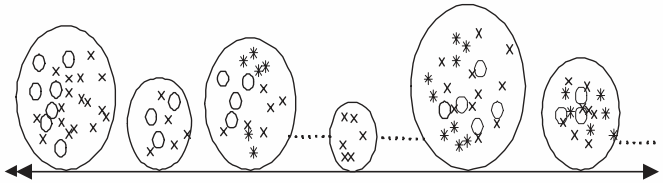
EIS



- Adaptive modelling of complex dynamic processes through incremental learning
- Evolving features
- Evolving models: evolving NN (ECOS) – DENFIS, EFuNN, evolving FS - eTS, EC, statistical learning (e,g. SVM), hybrid systems, quantum inspired EIS
- Extracting evolving rules – the rules that cause the process to evolve and that evolve themselves
- Facilitating discoveries across disciplines – Bioinformatics, Neuroinformatics, Health informatics, Robotics, Business, Environment

Online feature selection for EIS with incremental PCA and LDA

- Let us consider the case that the $(N+1)th$ training sample is presented. The addition of this new sample will lead to the changes in both of the mean vector and covariance matrix; therefore, the eigenvectors and eigenvalues should also be recalculated. The mean input vector is easily updated.

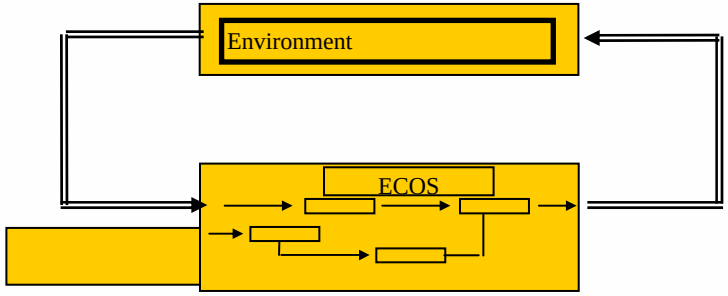


- If the new sample has almost all energy in the current eigenspace, dimensional augmentation is not needed. However, if it has some energy in the complementary space to the current eigenspace, the dimensional augmentation cannot be avoided. When the norm of the residue vector is larger than a threshold value, it must allow the number of dimensions to increase from k to $k+1$, and the current eigenspace must be expanded.
- Application: Face recognition
- Recent publications:

  – S. Ozawa, S.Too, S.Abe, S. Pang and N. Kasabov, *Incremental Learning of Feature Space and Classifier for Online Face Recognition*, Neural Networks, August, 2005
  – S. Pang, S. Ozawa and N. Kasabov, *Incremental Linear Discriminant Analysis for Classification of Data Streams*, IEEE Trans. SMC-B, vol. 35, No. 4, 2005

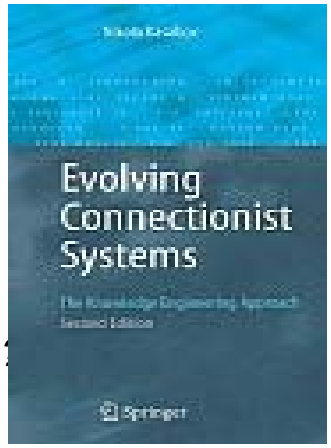
Evolving COnnectionist Systems – ECOS.

- ECOS are modular connectionist-based systems that evolve their structure and functionality in a continuous, self-organised, possibly on-line, adaptive, interactive way from incoming information; they can process both data and knowledge in a supervised and/or unsupervised way.



- Early examples of ECOS:
  - RAN (J.Platt, 1991) – evolving RBF NN
  - RAN with a long term memory – Abe et al, ;
  - Incremental FuzzyARTMAP;
  - Growing gas; etc.

- More recent developments:
  - EFuNN (Kasabov, 1998, 2001), DENFIS (Kasabov and Song, 2002)
  - EFuRS, eTS (P.Angelov, 2002)
  - SOFNN (McGinnity, Prasad, Leng, 2004)
  - TWNFI (Song and Kasabov, 2005)
  - Many other …..

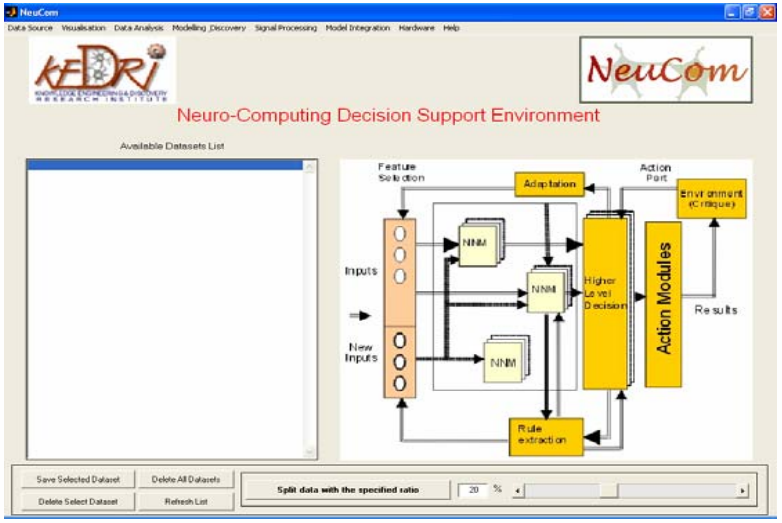


- '*Throw the "chemicals" and let the system grow, is that what you are talking about, Nik* *Walter Freeman, UC at Berkeley, a comment at "Iizuka""1998 conference*

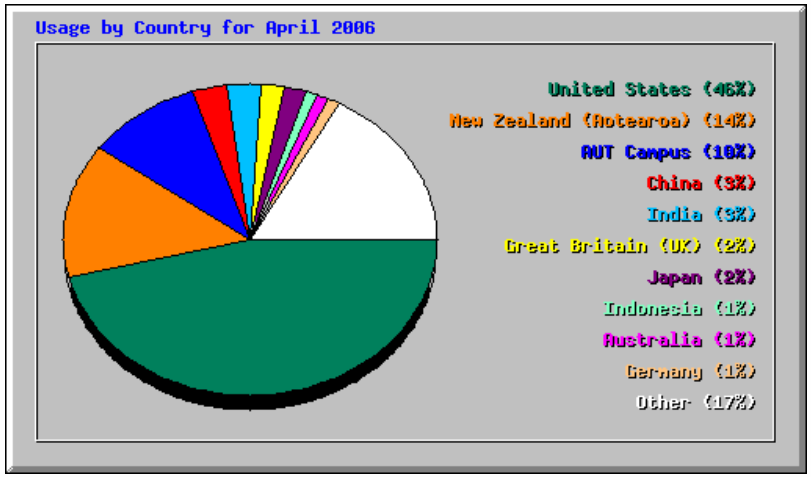
- N.Kasabov, Evolving connectionist systems: The knowledge engineering approach, second edition, Springer, 2007 (first edition 2002)

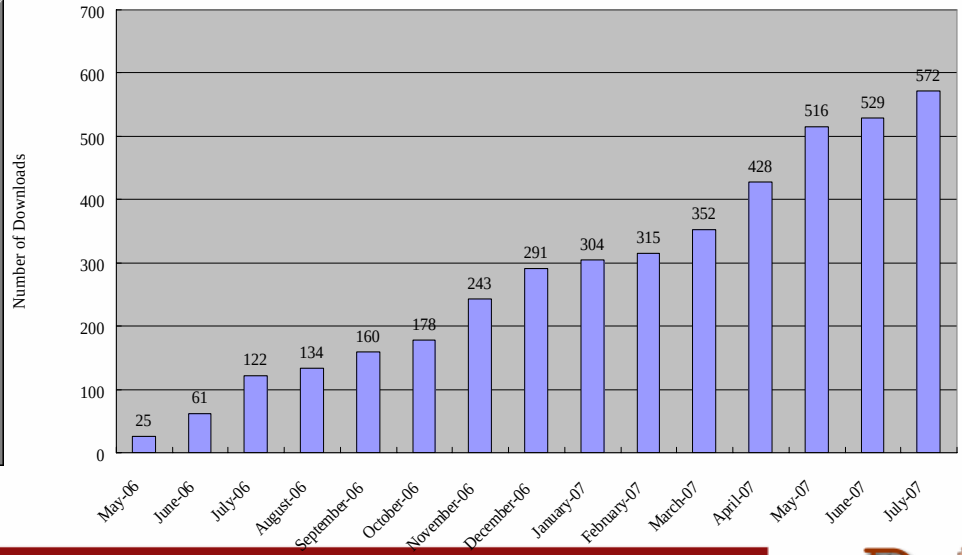
NeuCom

- **NeuCom** (www.theneucom.com): (P.Hwang et al.)
- A Software Environment for Data Mining and Intelligent Systems Design
- Incorporates 60 traditional and new techniques for intelligent data analysis and the creation of intelligent systems
- A free copy available for education and research from: www.theneucom.com
- Adopted in 70 research laboratories, institutes and universities from all over the world

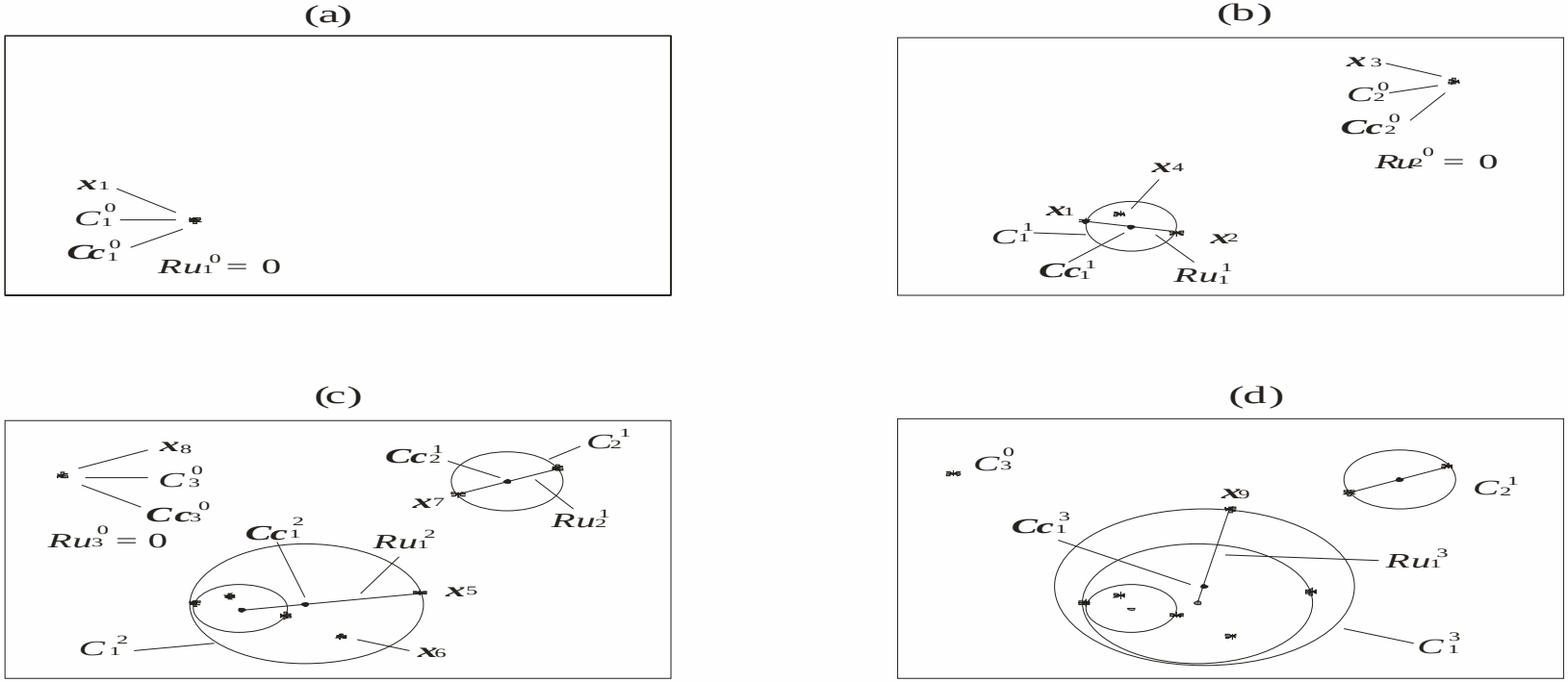


NeuCom Usage





2. Evolving Connectionist Methods for Unsupervised Learning: Clustering



- ECM: Fast one-pass (or at most - several passes) algorithm for dynamic clustering of a stream of data
- Performs a simple evolving, on-line, maximum distance based clustering
- The figure shows an evolving clustering process using ECM with consecutive examples x1 to x9 in a 2D space
- If the learning is supervised – a local function is evolved in each cluster
- Demo

Evolving Self-organsing Maps - ESOM

Step 1: Input a new data vector **x**.

Step 2: Find a set S of prototypes that are closer to **x** than a pre-defined threshold.

Step 3: If S is null, go to step 4 (insertion), otherwise – calculate the activations ai of all nodes from S and go to step 5 (updating).

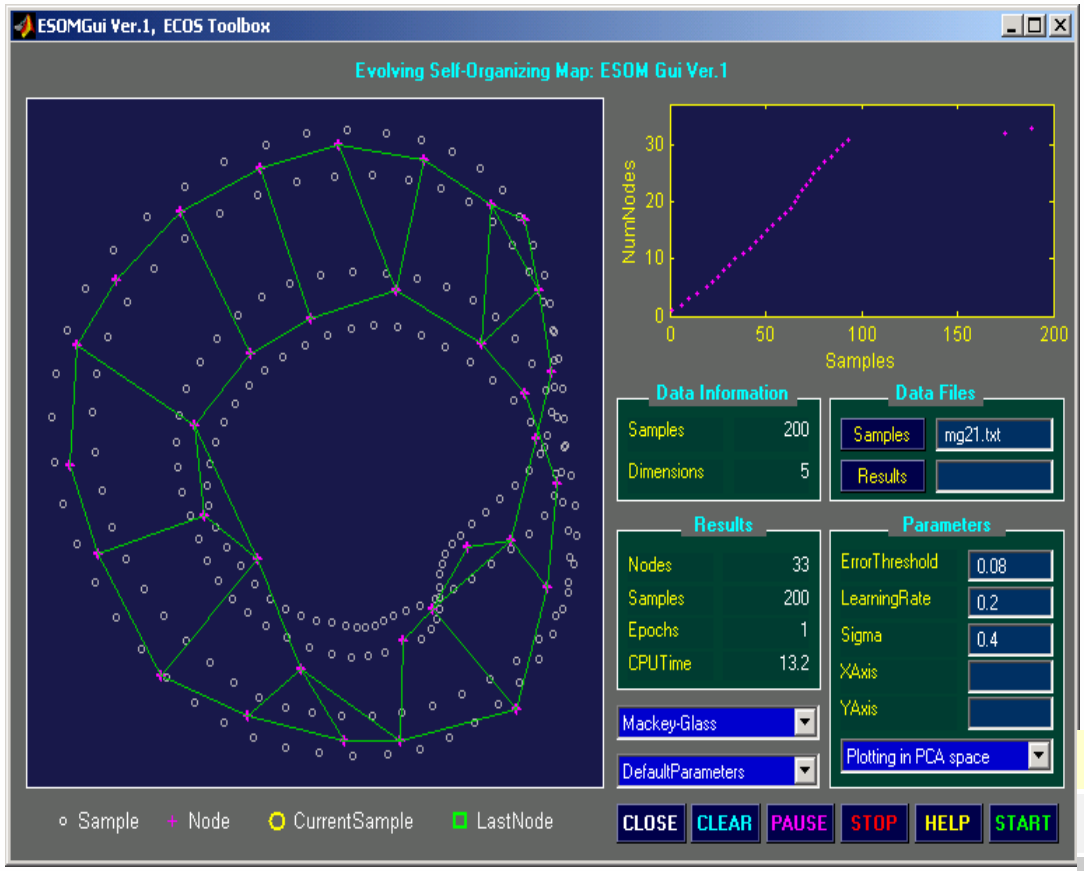
Step 4 (insertion): Create a new node **w**i for **x** and make a connection between this node and its two closest nodes (nearest neighbours) that will form a set S.

Step 5 (updating): Modify all prototypes in S according to (2.17) and re-calculate the connections s(i,j) between the winning node i (or the newly created one) and all the nodes j in the set S: s(i,j) = aiaj / max {ai, aj}

Step 6: After a certain number of input data are presented to the system, prune the weakest connections. If isolated nodes appear, prune them as well.

Step 7: go to step 1

ESOM examples



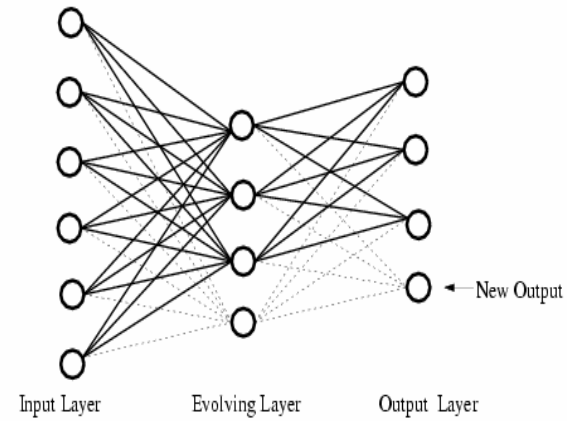
Model	No. of units	Error rate	No. of Epochs
GCS	145	0	180
DCS-GCS	135	0	135
LVQ	114	11.9%	50
SOM	144	22.2%	50
ESOM	105	0	1

3. Evolving Connectionist Methods for Supervised Learning

- Hidden nodes evolve, starting from no nodes at all.
- Each hidden node is a cluster center.
- Clusters grow in radius and shrink through a learning algorithm
- Each hidden node represents a local model (a rule) that associates an input cluster with an output function, e.g. a constant label, a linear function, a non-linear function, etc
- If a new input vector belongs to a cluster to certain degree, than the corresponding local model applies, otherwise – m of the closest models are used to calculate the output.
- Incremental **supervised clustering** with new input vectors **x**
- First layer of connections: $W1(r_j(t+1))=W1(r_j(t)) + lj.\ D(\mathbf{x}, W1(r_j(t)))$
- Second layer: $W2(r_j(t+1)) = W2(r_j(t)) + lj.\ (\mathbf{y} - A2).\ A1(r_j(t))$,

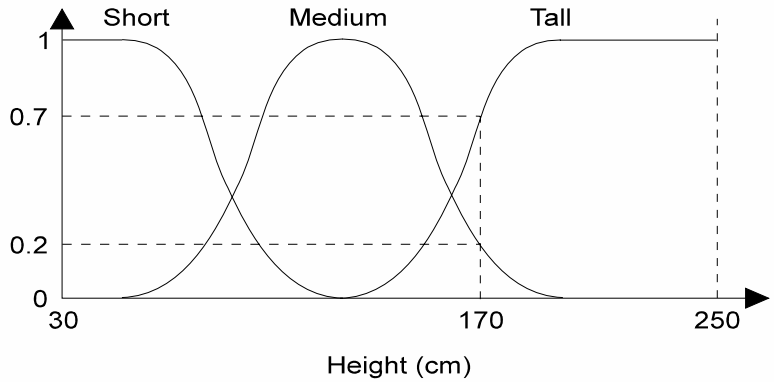
where: r_j is the jth rule node (hidden node); D – distance; A2=f2(W2.A1) is the activation vector of the output neurons when **x** is presented;

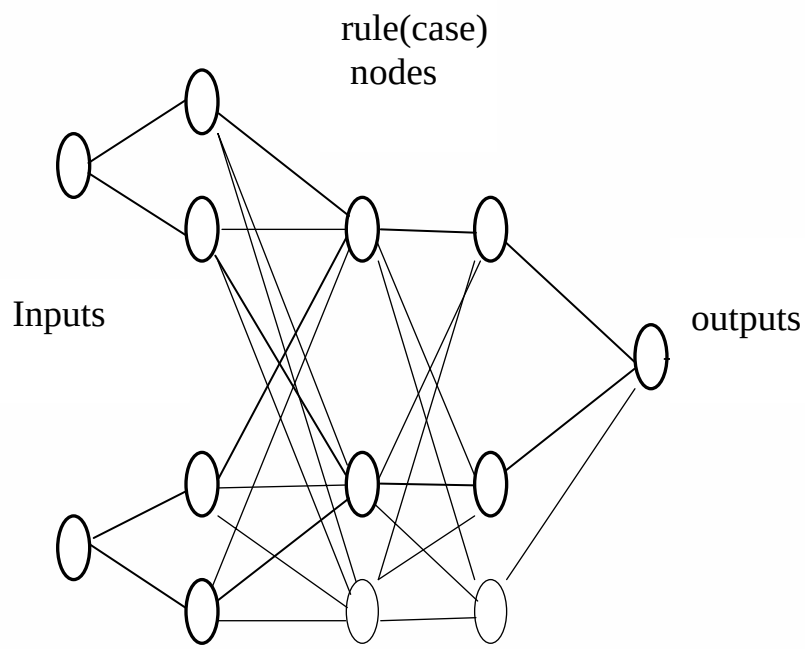
- $A1(r_j(t)) = f2(D(W1(r_j(t)), \mathbf{x}))$ is the activation of the rule node $r_j(t)$;
- a simple linear function can be used for f1 and f2, e.g. $A1(rj(t)) = 1 - D(W1(r_j(t)), \mathbf{x})$;
- lj is the current learning rate of the rule node r_j calculated for example as $lj = 1/Nex(r_j)$, where $Nex(r_j)$ is the number of examples associated with rule node r_j.



Evolving fuzzy neural networks EFuNN and ECF

- As a general case, input and/or output variables can be non-fuzzy (crisp) or fuzzy
- Fuzzy variables
- Example of three Gaussian MF





- *EFuNN, N. Kasabov, IEEE Tr SMC, 2001*
- Partial case: ECF – evolving classifier function – no output MF, only input MF.
- Supervised clustering
- Zadeh-Mamdani fuzzy rules
- Simple version – **ECF.** Parameters: Rmax, Rmin, #input MF (e.g. 1,2,3,...), m-of-n (e.g. 1,2,3,..), # iterations for training (e.g. 1,2,3, …
- Demo

The ECM Algorithm

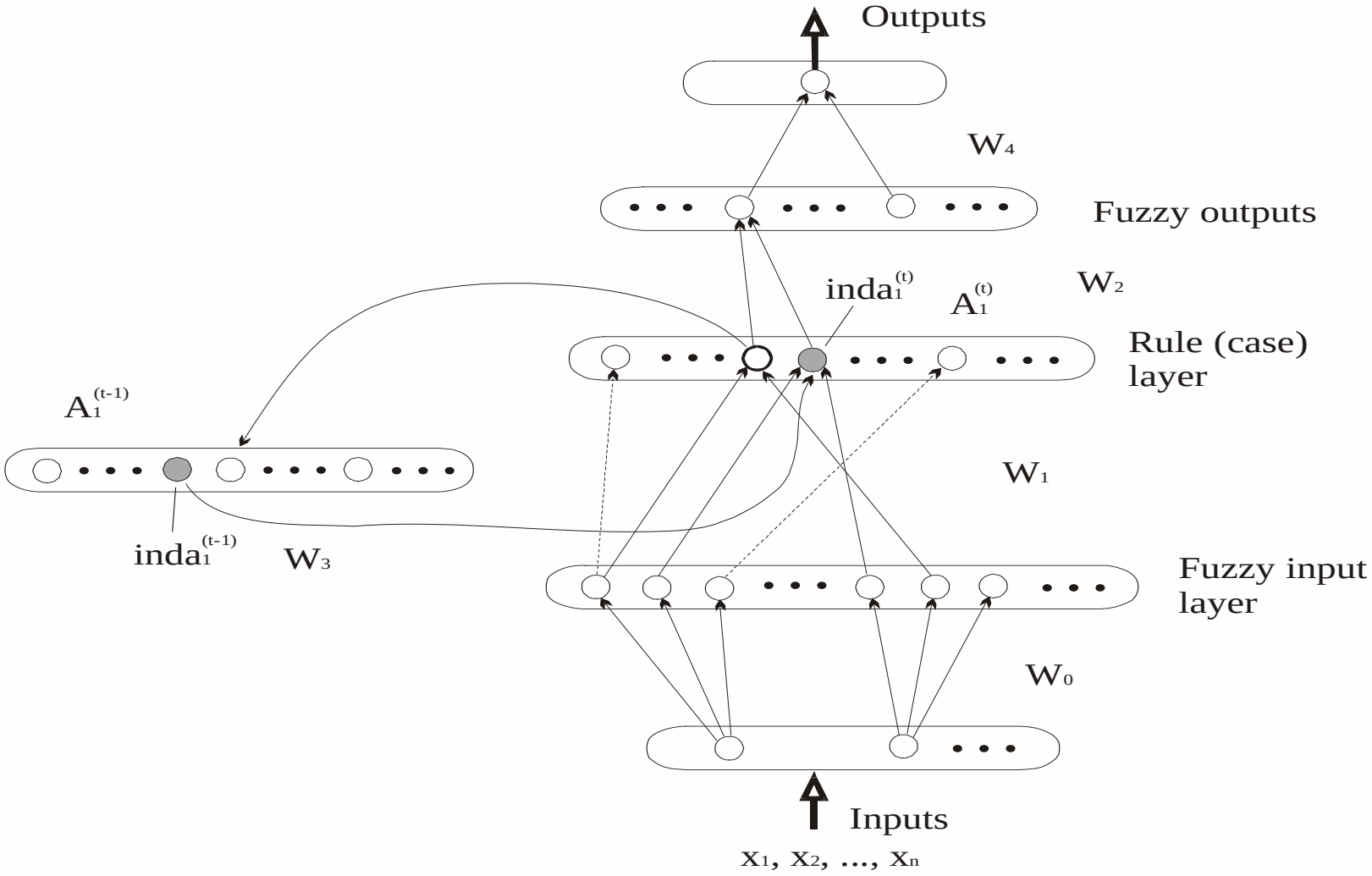
Learning algorithm of ECF:

1. Enter the current input vector from the data set (stream) and calculate the distances between this vector and all rule nodes already created using Euclidean distance (by default). If there is no node created, create the first one that has the coordinates of the first input vector attached as input connection weights.
2. If all calculated distances between the new input vector and the existing rule nodes are greater than a max-radius parameter Rmax, a new rule node is created. The position of the new rule node is the same as the current vector in the input data space and the radius of its receptive field is set to the min-radius parameter Rmin; the algorithm goes to step 1; otherwise it goes to the next step.
3. If there is a rule node with a distance to the current input vector less then or equal to its radius and its class is the same as the class of the new vector, nothing will be changed; go to step 1; otherwise:
4. If there is a rule node with a distance to the input vector less then or equal to its radius and its class is different from those of the input vector, its influence field is reduced. The radius of the new field is set to the larger value from the two numbers: distance minus the min-radius; min-radius. New node is created as in 2 to represent the new data vector.
5. If there is a rule node with a distance to the input vector less than or equal to the max-radius, and its class is the same as of the input vector's, enlarge the influence field by taking the distance as a new radius if only such enlarged field does not cover any other rule nodes which belong to a different class; otherwise, create a new rule node in the same way as in step 2, and go to step 1.

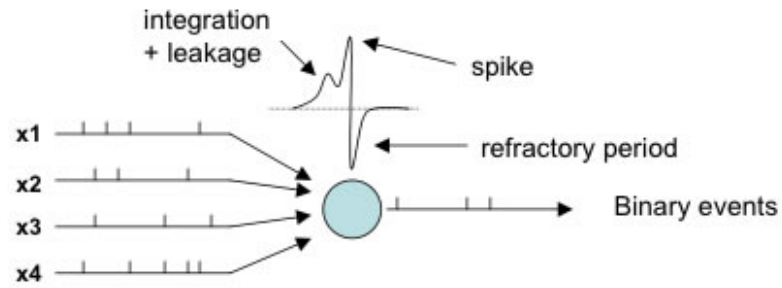
Recall procedure (classification of a new input vector) in a trained ECF :

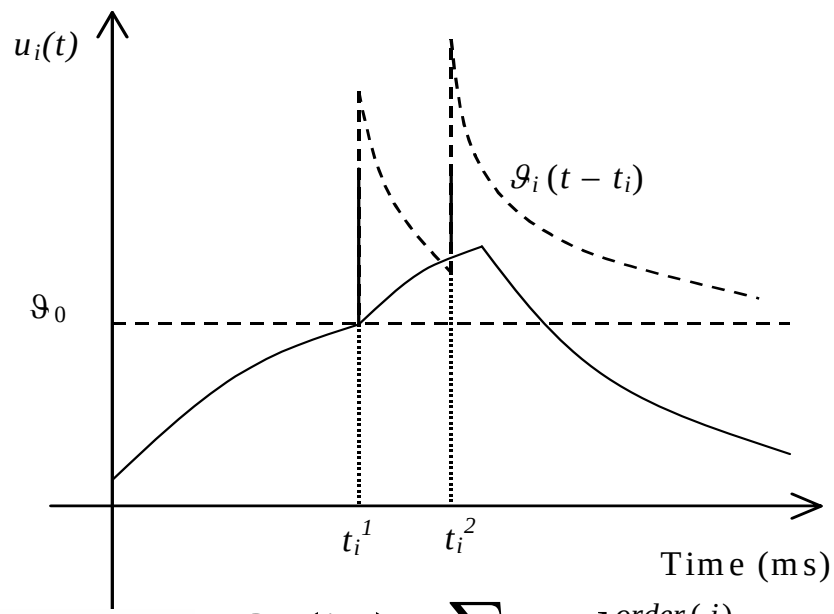
1. Enter the new vector in the ECF trained system; If the new input vector lies within the field of one or more rule nodes associated with one class, the vector is classified in this class;
2. If the input vector lies within the fields of two or more rule nodes associated with different classes, the vector will belong to the class corresponding to the closest rule node.
3. If the input vector does not lie within any field, then take m highest activated by the new vector rule nodes, and calculate the average distances from the vector to the nodes with the same class; the vector will belong to the class corresponding to the smallest average distance.

An EFuNN architecture with a short term memory and feedback connections



4. Brain like ECOS: Evolving spiking neural networks





$$PSP(i,t) = \sum \mathrm{mod}^{order(j)} w_{j,i}$$

ESNN

Evolving SNN (ESNN) evolve/develop their structure and functionality in an incremental way from incoming data based on the following principles (Gerstner and Kistler 2002; Wysoski, Benuskova et al. 2006; Benuskova and Kasabov 2007; Kasabov 2007):

- New spiking neurons are created to accommodate new data, e.g. new output classes, such as faces in a face recognition system.
- Spiking neurons are merged if they represent the same concept (class) and have similar connection weights.

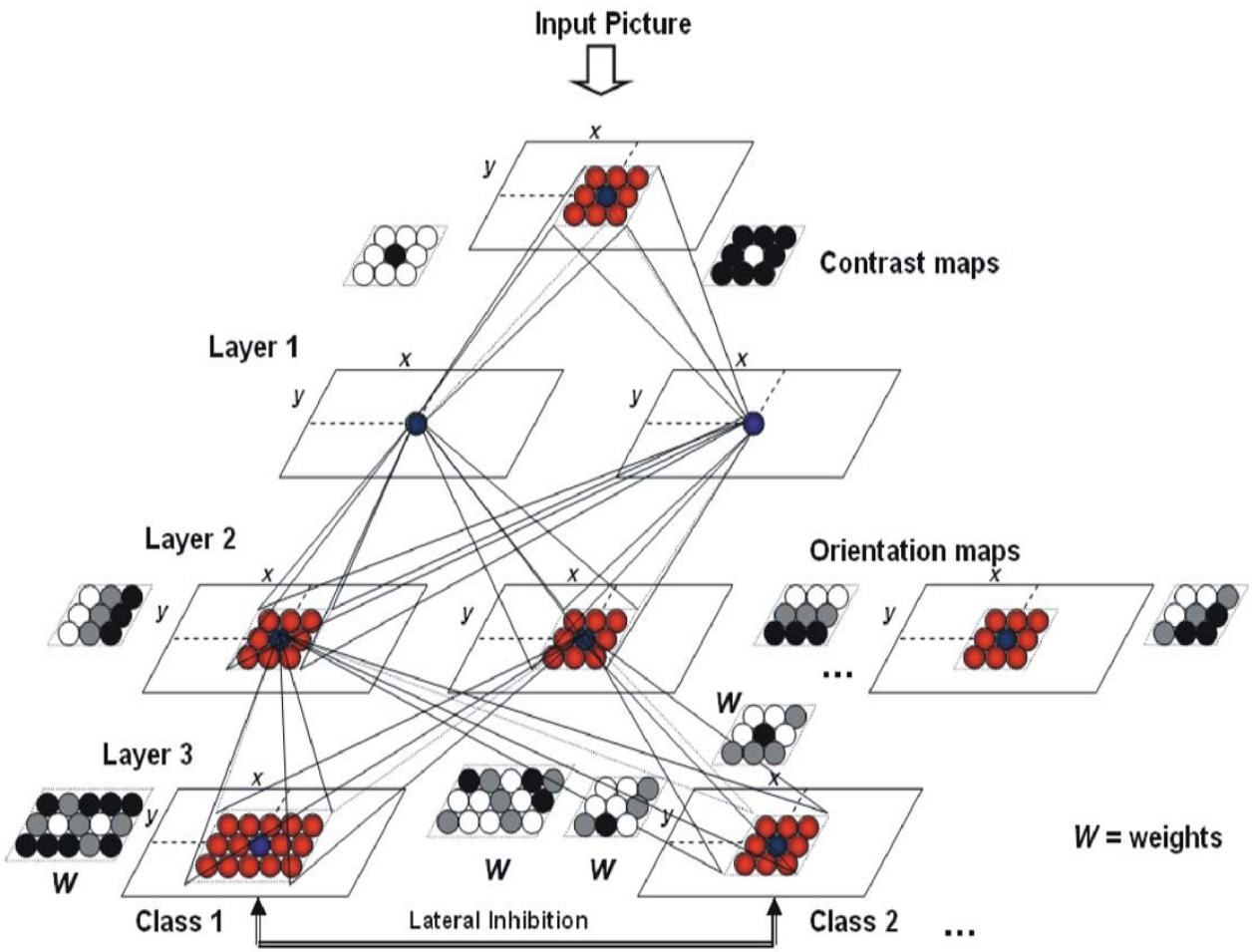
A general algorithm for ESNN:

For each input vector:

- *Create (evolve) a new spiking neuron*
- *Propagate the input into the network*
- *Train the newly created neuron using proposed equations*
- *Calculate the similarity between weight vectors of newly created neuron and existent neurons within the neuronal map*
- *IF similarity > Threshold, THEN merge the newly created neuron with the most similar neuron using proposed equations*

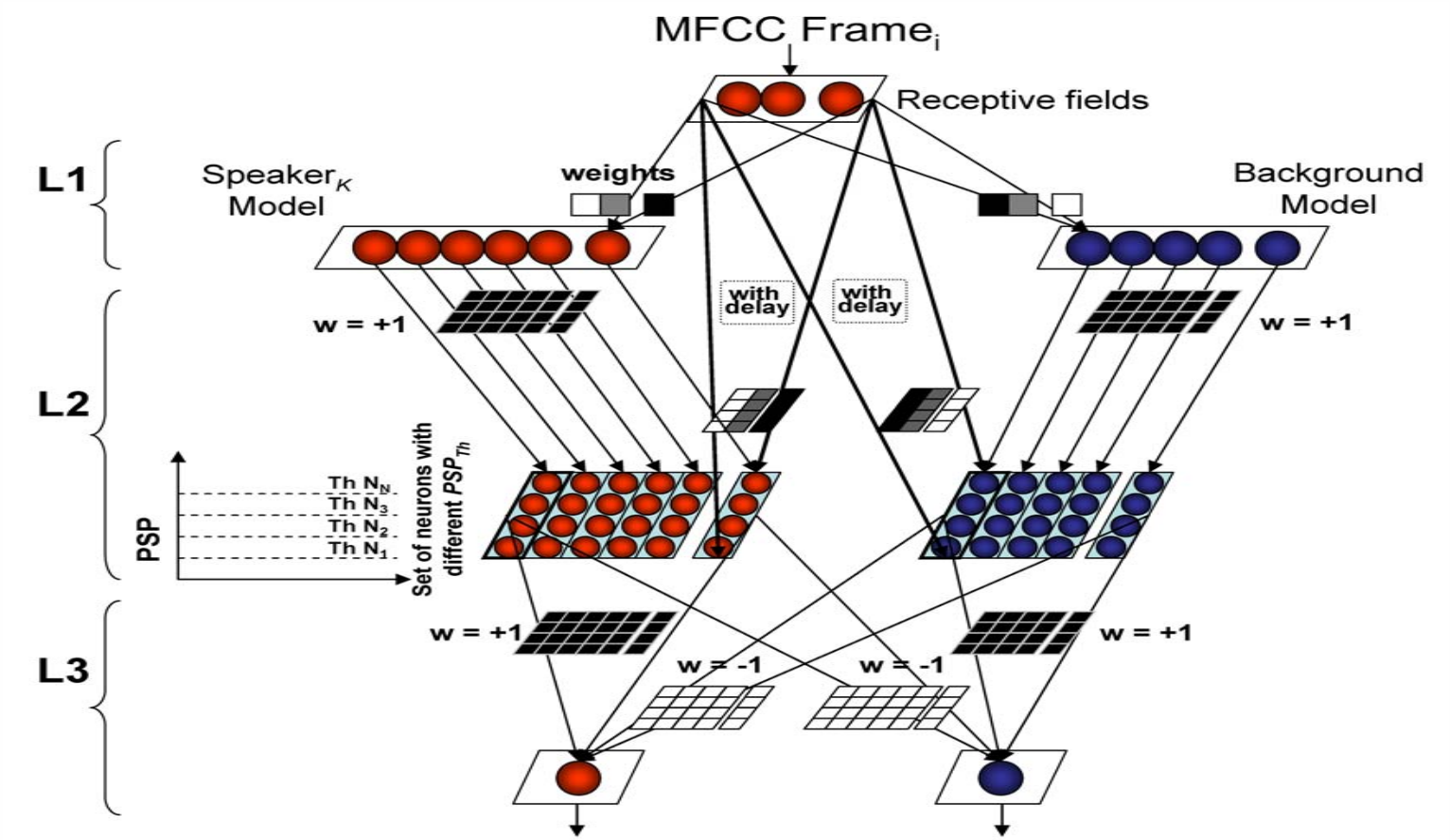
Evolving SNN for Image/Face Recognition

(Wysoski, Benuskova, Kasabov, Proc. ICANN 2006



Evolving SNN for Speaker Identification

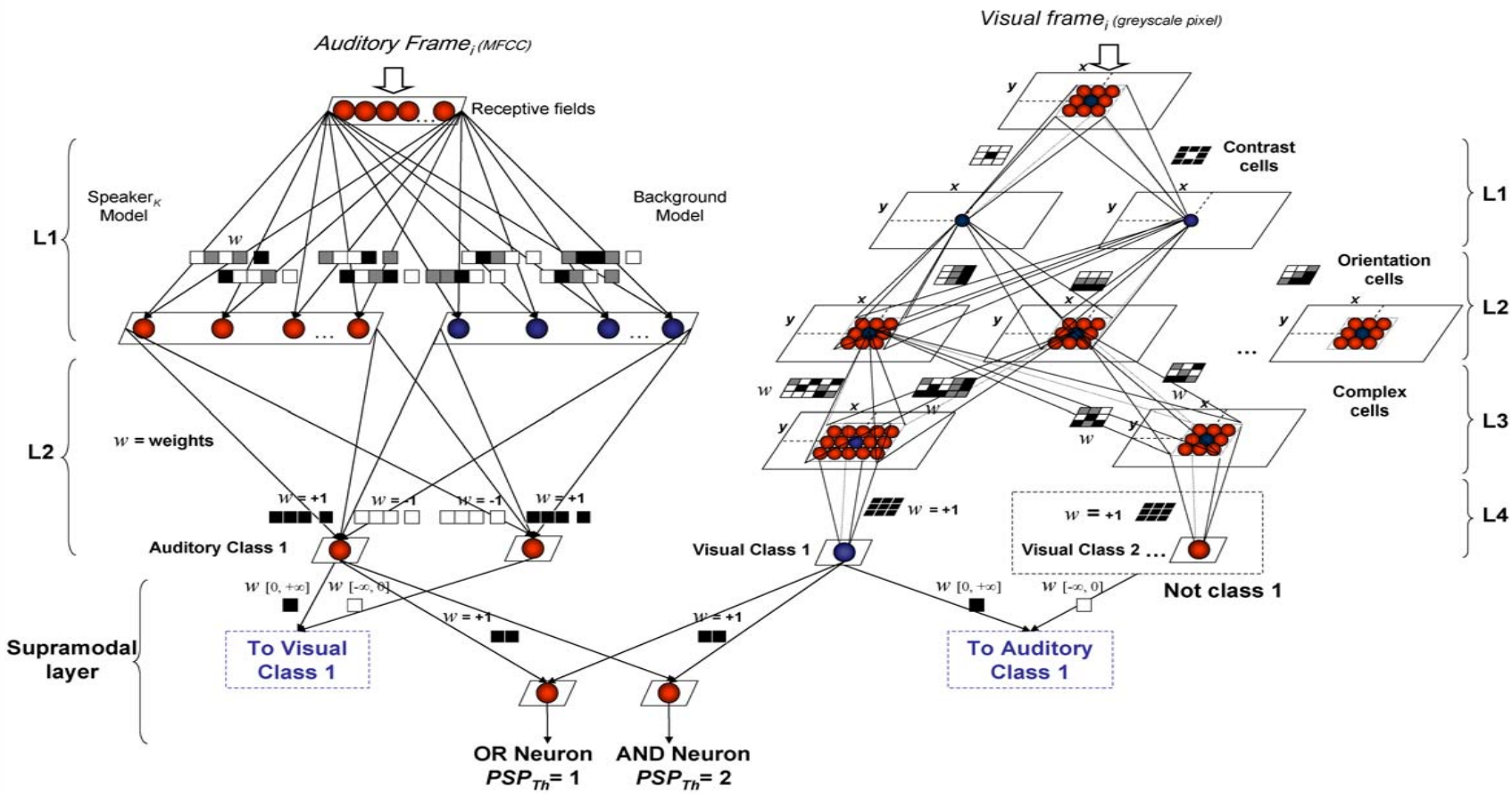
(Wysoski, Benuskova, Kasabov, Poster session, Monday afternoon, Proc. ICANN 2007



Evolving SNN for Audiovisual Pattern Recognition

Person identification based on speech and face data

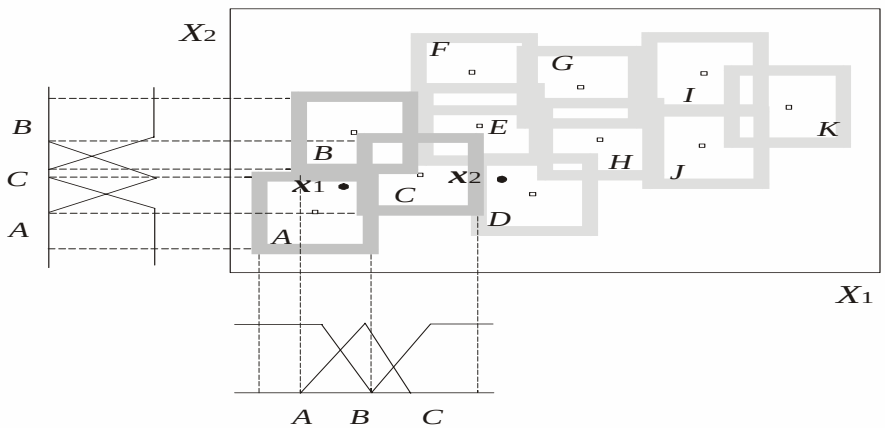
(Wysoski, Benuskova, Kasabov, Proc. ICONIP, 2007



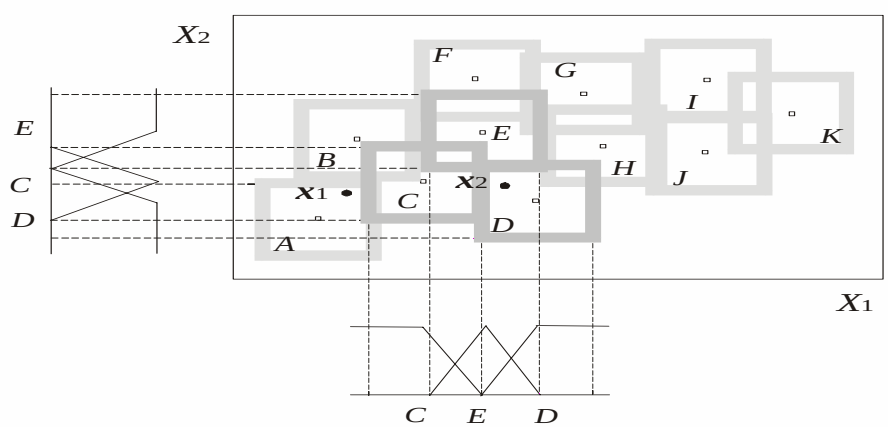
5. Evolving Neuro-Fuzzy Inference Models

(DENFIS, Kasabov and Song, 2002, IEEE Tr Fuzzy Systems)

(a) Fuzzy rule group 1 for a DENFIS



(b) Fuzzy rule group 2 for a DENFIS



DENFIS algorithm:

(1) Learning:

- Unsupervised, incremental clustering.
- For each cluster there is a Takagi-Sugeno fuzzy rule created: IF x is in cluster Cj THEN yj = fj (x),

where: $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_q$

- Incremental learning of the function coefficients and weights of the functions through least square error

(2) Fuzzy inference over fuzzy rules:

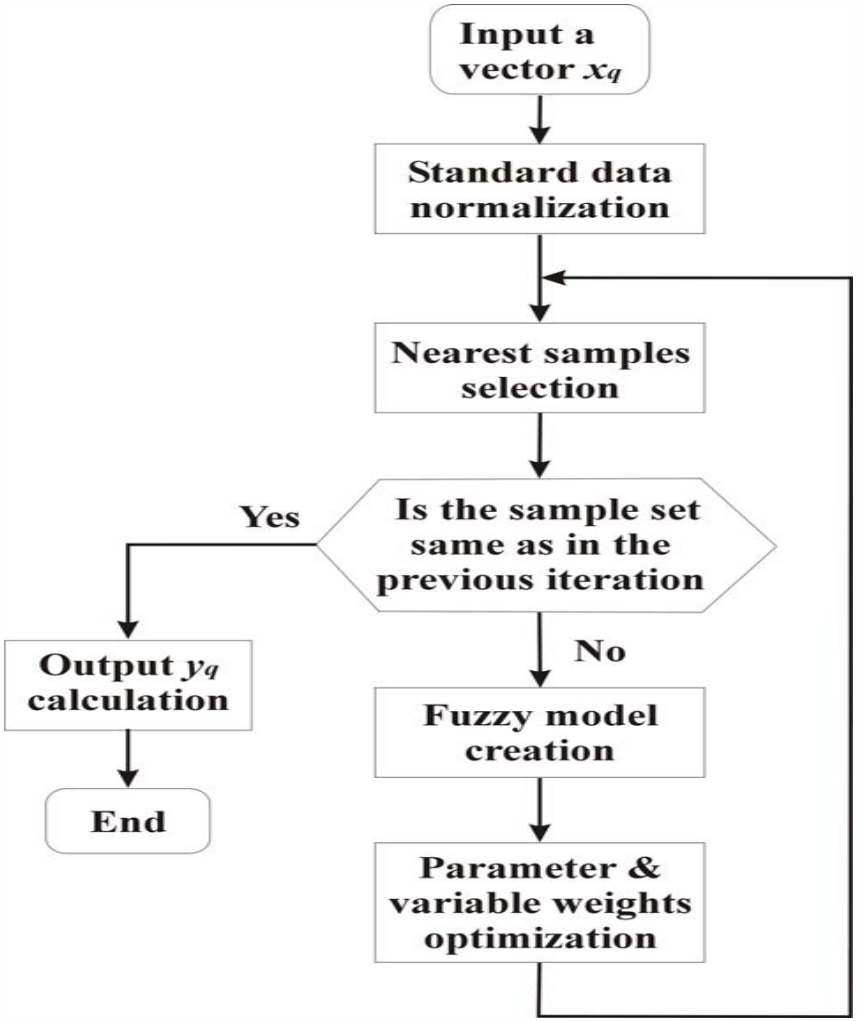
- For a new input vector x = [x1,x2, … , xq] DENFIS chooses m fuzzy rules from the whole fuzzy rule set for forming a current inference system.
- The inference result is:

$$y = \frac{\sum_{i=1,m} [\omega_i f_i(x_1, x_2, \ldots, x_q)]}{\sum_{i=1,m} \omega_i}$$

Two fuzzy rule groups are formed by DENFIS to perform inference for 2 input vectors

Transductive Neuro Fuzzy Inference with Weighted Data Normalisation – TWNFI, for personalised modelling

(Q.Song and N.Kasabov, IEEE Tr FS, December 2005, and Neural Networks, Dec. 2006)



After the nearest samples are selected for an input vector **x,** the samples are clustered using ECM.

Fuzzy rules are created/derived for each cluster:

R_l :If x_1 is F_{l1} and x_2 is F_{l2} and … x_P is F_{lP}, then y is G_l,

where F_{lj} and Gl are fuzzy sets defined by Gaussian type membership functions.

Input variable weights w_j and fuzzy rule parameters are optimized through the steepest descent algorithm.

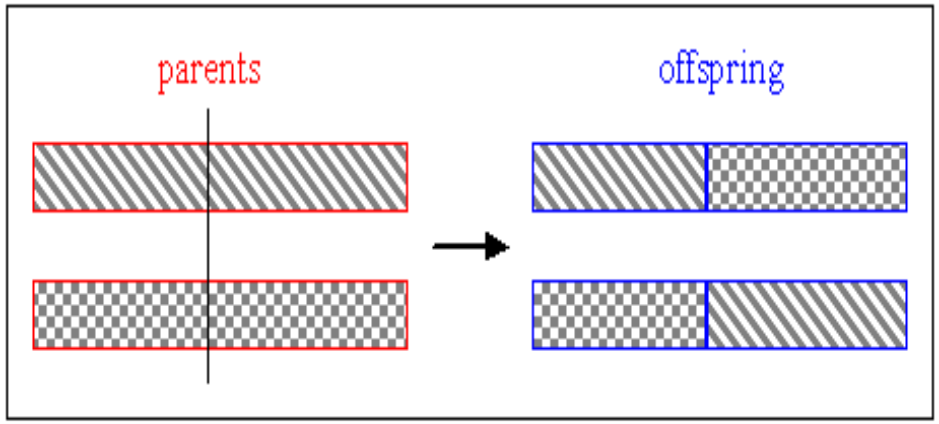
$$f(\boldsymbol{x_i}) = \frac{\sum_{l=1}^{M} \frac{n_l}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left[-\frac{w_j^2 (x_{ij} - m_{lj})^2}{2\sigma_{lj}^2}\right]}{\sum_{l=1}^{M} \frac{1}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left[-\frac{w_j^2 (x_{ij} - m_{lj})^2}{2\sigma_{lj}^2}\right]}$$

6. Population-Generation Based methods: Evolutionary Computation for Parameter and Structure optimisation of ECOS

Evolutionary computation.

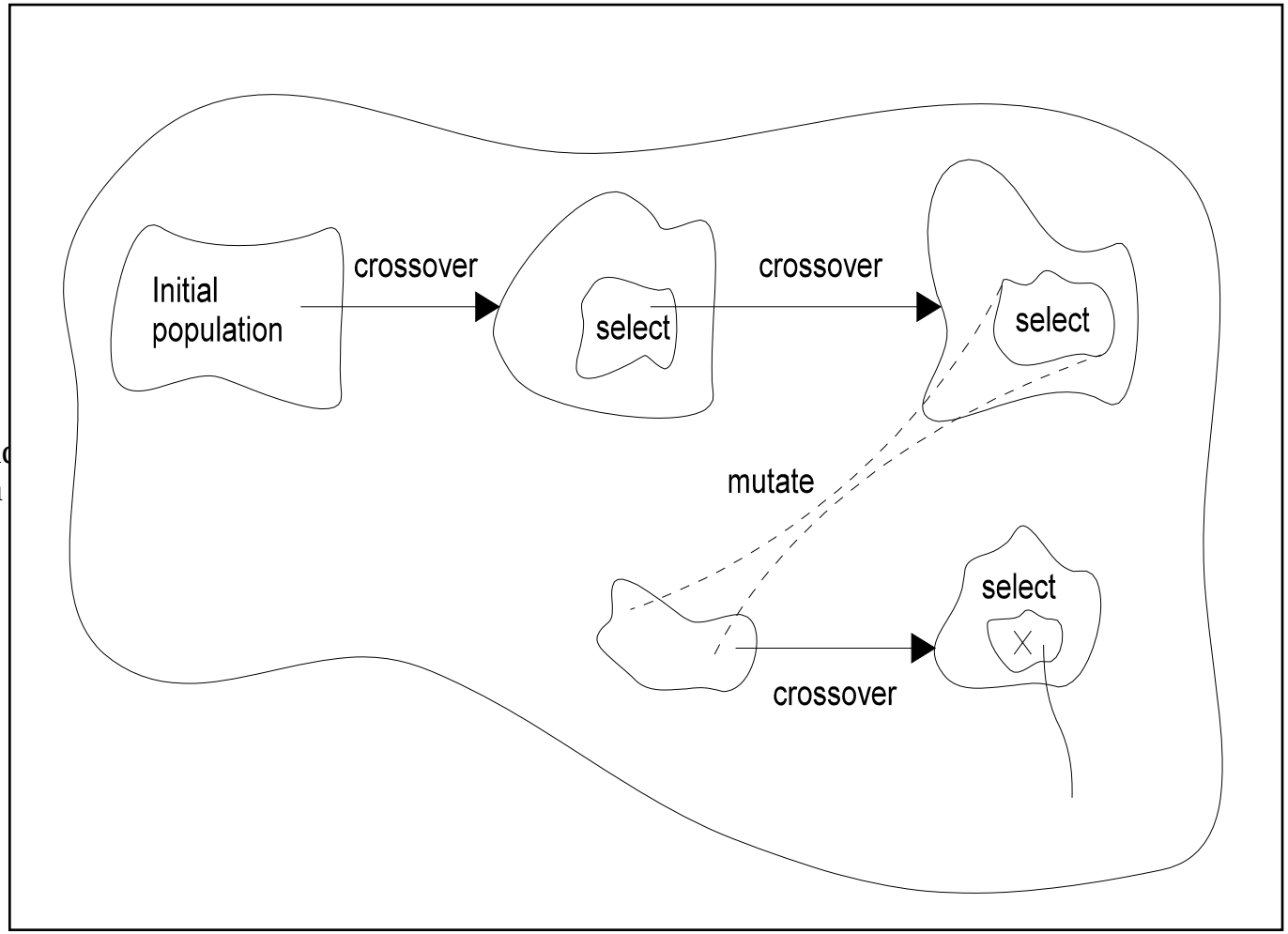
Terminology:

- *Gene*
- *Chromosome*
- *Population*
- *Crossover*
- *Mutation*
- *Fitness function*
- *Selection*



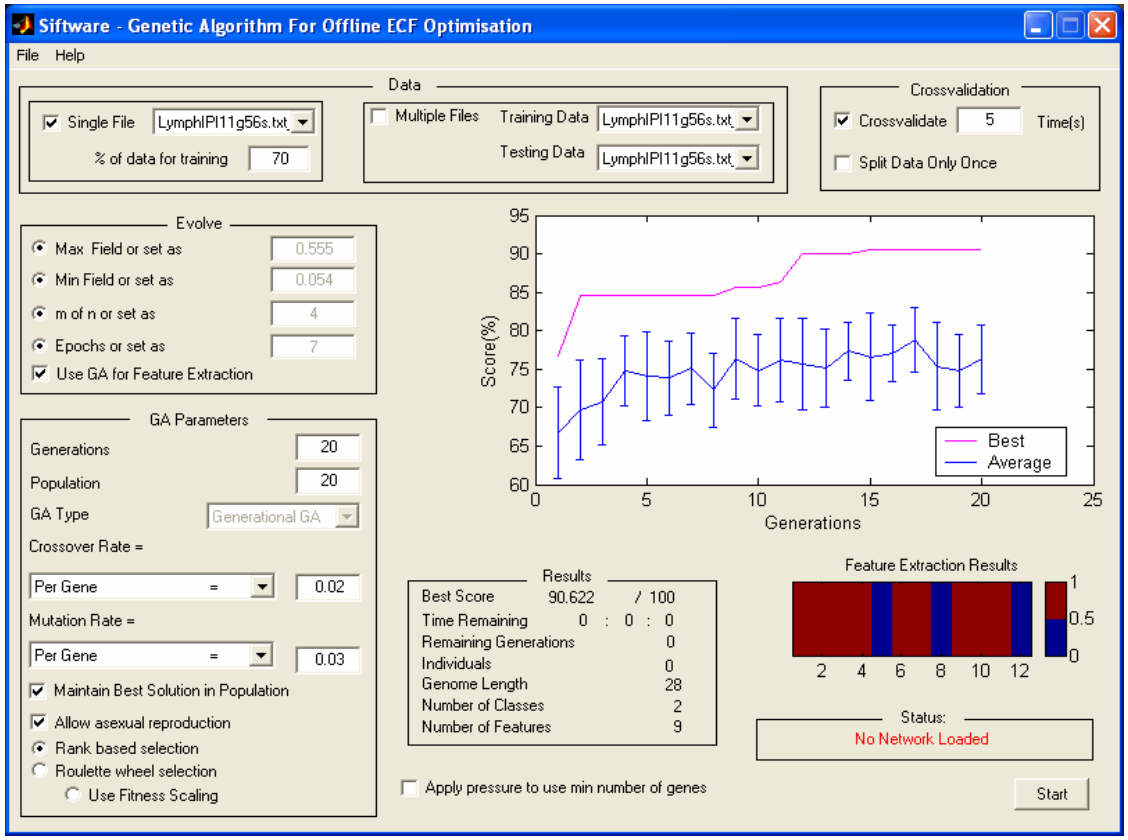
GA and ECOS

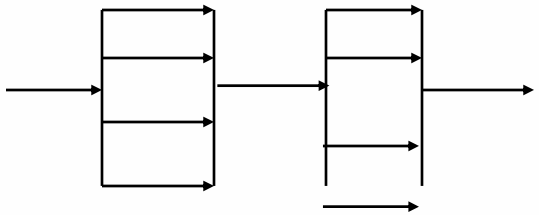
- Many individual ECOS are evolved simultaneously on the same data through a GA method
- A chromosome represents each individual ECOS parameters
- Individuals are evaluated and the best one is selected for a further development
- Mutation



EC for feature, parameter and structure optimisation of ECF ECOS

- GA optimisation of the parameters of the model and the input features
- A chromosome contains as “genes” all model parameters and input features (yes, no)
- Replication of individual models and selection of:
  - The best one
  - The best *m* averaged, etc





7. Evolving Integrated Multimodel Systems

Model and data integration through ECOS

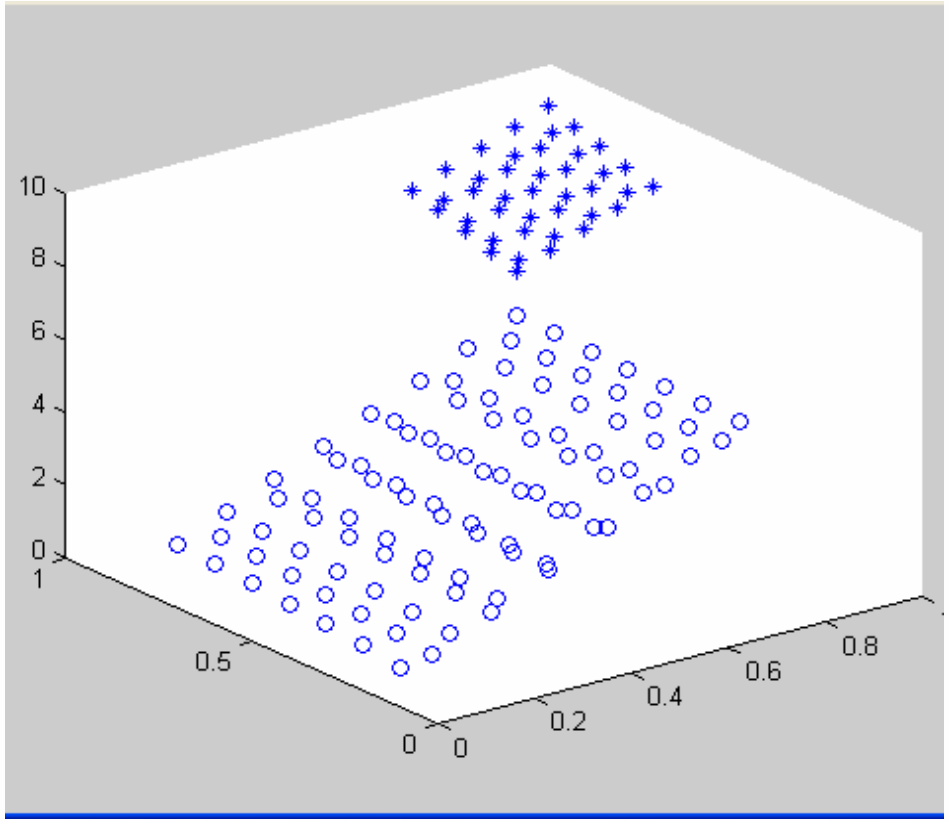
-**Inserting initial rules** (existing knowledge) and training with new data

- **Generating data from existing model** and training an ECOS on both old and new data

- **New rules evolve continuously**

Example: A 3D plot of data D_0 (data samples denoted as "o") generated from a model M (formula) $y = 5.1x_1 + 0.345x_1^2 - 0.83x_1 \log_{10} x_2 + 0.45x_2 + 0.57 \exp(x_2^{0.2})$ in the sub-space of the problem space defined by x_1 and x_2 both having values between 0 and 0.7, and

• New data D (samples denoted as "*") defined by x_1 and x_2 having values between 0.7 and 1;



Prototype rules evolved through DENFIS and EFuNN after model and new data integration

Takagi-Sugeno fuzzy rules (DENFIS):

- Rule 1: IF x_1 is (-0.05, 0.05, 0.14) and x_2 is (0.15,0.25,0.35) THEN $y = 0.01 + 0.7x_1 + 0.12x_2$
- Rule 2: IF x_1 is (0.02, 0.11, 0.21) and x_2 is (0.45,0.55, 0.65) THEN $y = 0.03 + 0.67x_1 + 0.09\ _{x2}$
- Rule 3: IF x_1 is (0.07, 0.17, 0.27) and x_2 is (0.08,0.18,0.28) THEN $y = 0.01 + 0.71x_1 + 0.11x_2$
- Rule 4: IF $x_1$is (0.26, 0.36, 0.46) and x_2 is (0.44,0.53,0.63) THEN $y = 0.03 + 0.68x_1 + 0.07x_2$
- Rule 5: IF $x_1$is (0.35, 0.45, 0.55) and x_2 is (0.08,0.18,0.28) THEN $y = 0.02 + 0.73x_1 + 0.06x_2$
- Rule 6: IF $x_1$is (0.52, 0.62, 0.72) and x_2 is (0.45,0.55,0.65) THEN $y = -0.21 + 0.95x_1 + 0.28x_2$
- Rule 7: IF $x_1$is (0.60, 0.69,0.79) and x_2 is (0.10,0.20,0.30) THEN $y = 0.01 + 0.75x_1 + 0.03x_2$

- **New rules:**
- **Rule 8: IF $x_1$is (0.65,0.75,0.85) and x_2 is (0.70,0.80,0.90) THEN $y = -0.22 + 0.75x_1 + 0.51x_2$**
- **Rule 9: IF $x_1$is (0.86,0.95,1.05) and x_2 is (0.71,0.81,0.91) THEN $y = 0.03 + 0.59x_1 + 0.37x_2$**

Zhade-Mamdani fuzzy rules (ECF, EFuNN):

Rule 1: IF x_1 is (Low 0.8) and x_2 is (Low 0.8) THEN y is (Low 0.8), radius R_1=0.24; N_{1ex}= 6

Rule 2: IF x_1 is (Low 0.8) and x_2 is (Medium 0.7) THEN y is (Small 0.7), R_2=0.26, N_{2ex}= 9

Rule 3: IF x_1 is (Medium 0.7) and x_2 is (Medium 0.6) THEN y is (Medium 0.6), R_3 = 0.17,N_{3ex}=17

Rule 4: IF x_1 is (Medium 0.9) and x_2 is (Medium 0.7) THEN y is (Medium 0.9), R_4 = 0.08, N_{4ex}=10

Rule 5: IF x_1 is (Medium 0.8) and x_2 is (Low 0.6) THEN y is (Medium 0.9), R_5= 0.1, N_{5ex} = 11

Rule 6: IF x_1 is (Medium 0.5) and x_2 is (Medium 0.7) THEN y is (Medium 0.7), R_6= 0.07,N_{6ex}= 5

New rules:

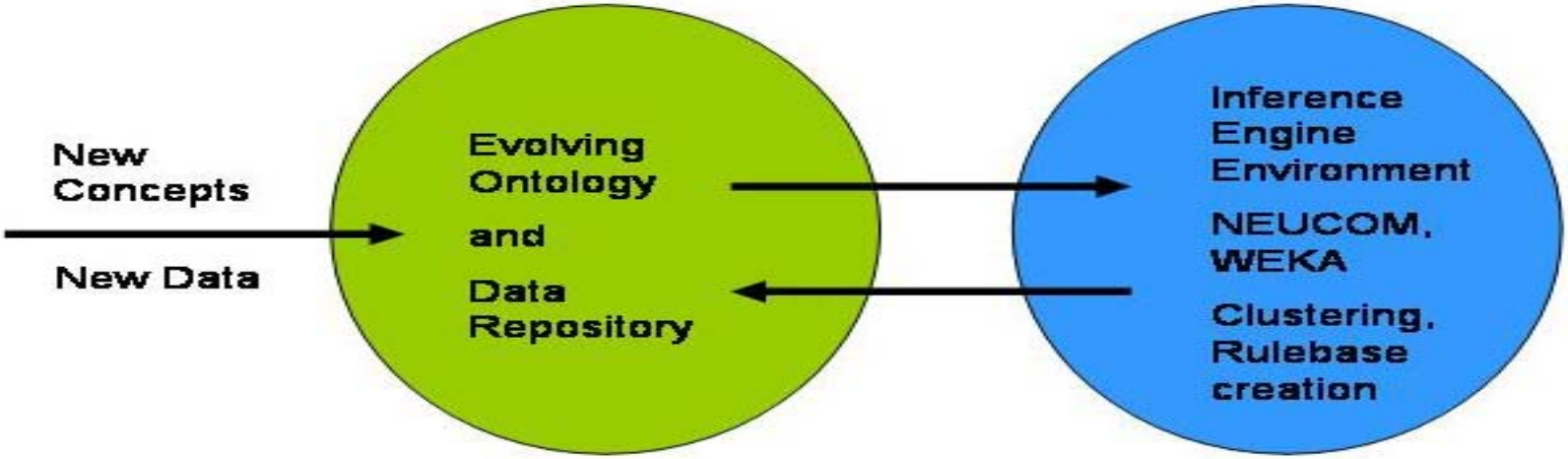
Rule 7: IF x1 is (High 0.6) and x2 is (High 0.7) THEN y is (High 0.6), R_7 = 0.2, N_{7ex} = 12

Rule 8: IF x1 is (High 0.8) and x2 is (Medium 0.6) THEN y is (High 0.6), R_8=0.1,N_{8ex}= 5

Rule 9: IF x1 is (High 0.8) and x2 is (High 0.8) THEN y is (High3 0.8), R_9= 0.1, N_{9ex} = 6

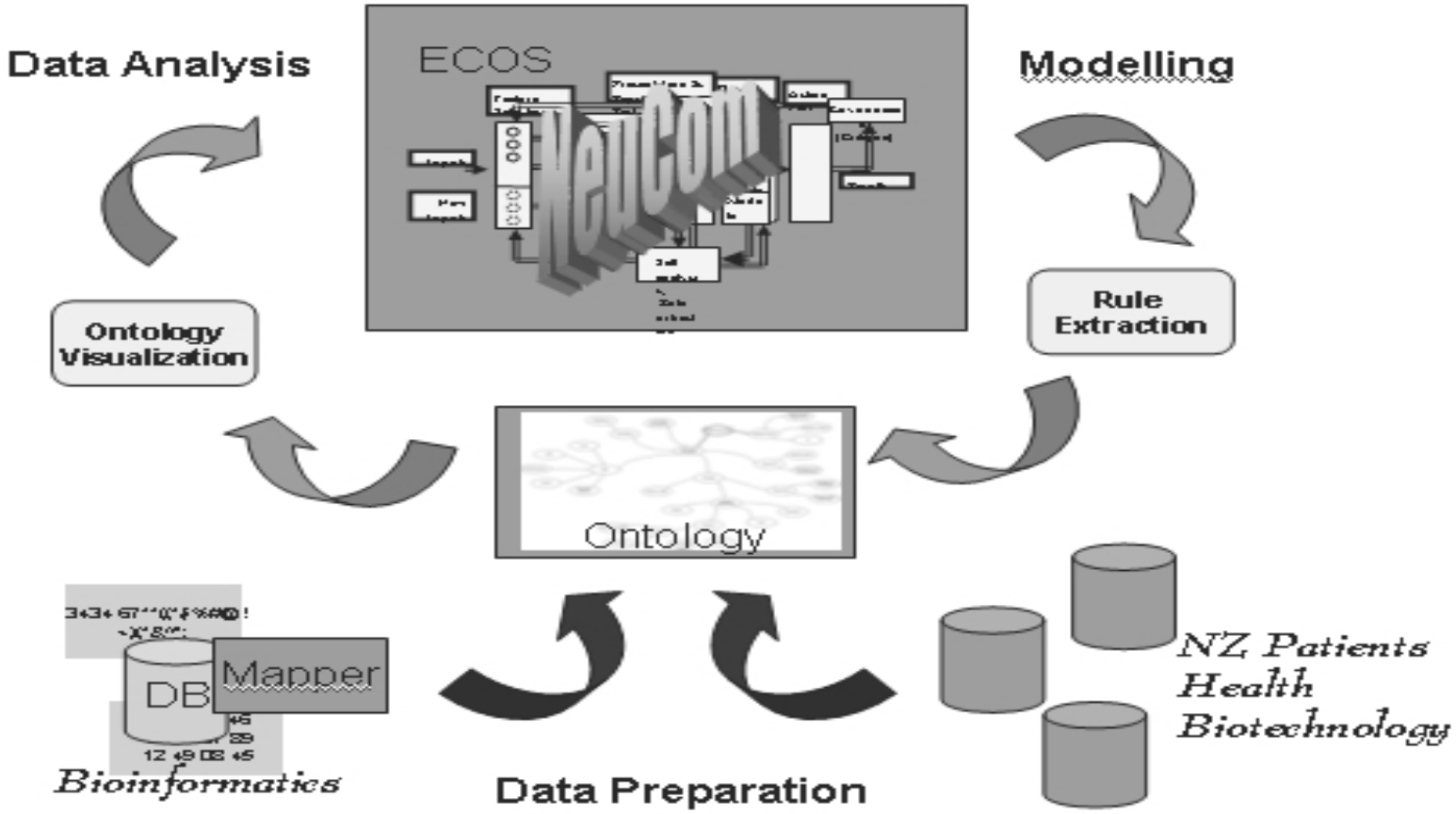
Integrating Ontology Systems and ECOS

- Ontology is a description (like a formal specification of a program) of the concepts and relationships between them that are important in a particular domain to support sharing and reuse of formally represented knowledge among users and systems
- Software development environments, e.g. Protégé, OWL,…
- Use for research, learning and teaching
- Existing ontologies, e.g. GO, have very simple information processing and search engines and are not “coupled” with computational intelligence



Evolving Ontology–based DSS

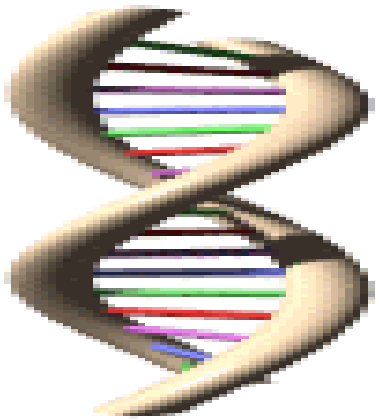
Gottgtroy P., Kasabov N., Macdonell S., Evolving Ontologies for Intelligent Decision Support, Elsevier, Fuzzy Logic And The Semantic Web, Chapter 21, pp 415-439, 2006

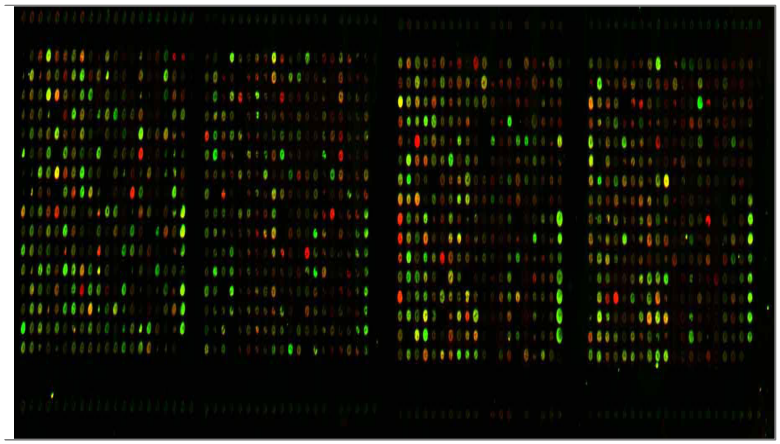


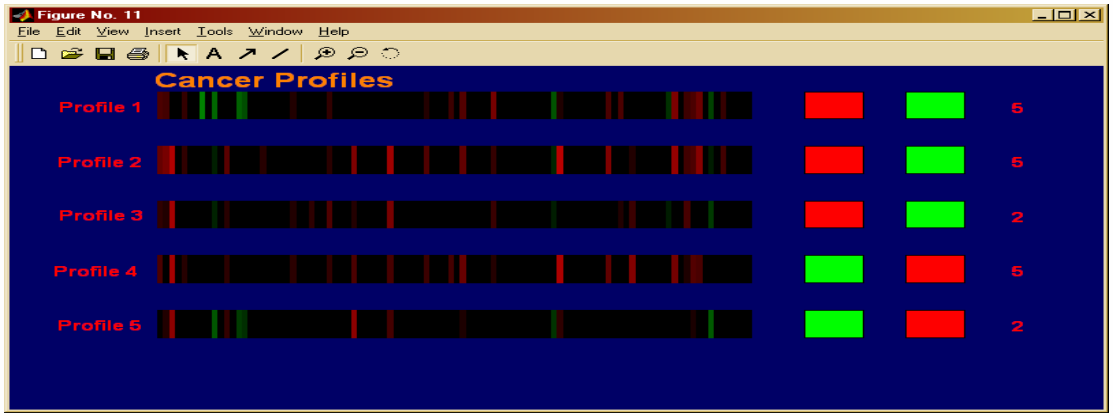
8. Bioinformatics

Gene expression data profiling

- DNA analysis - large data bases; data always being added and modified; different sources of information
- Markers and drug discoveries:
  - Gastric cancer
  - Bladder cancer
  - CRC
  - www.pedblnz.com
- Specialised software SIFTWARE



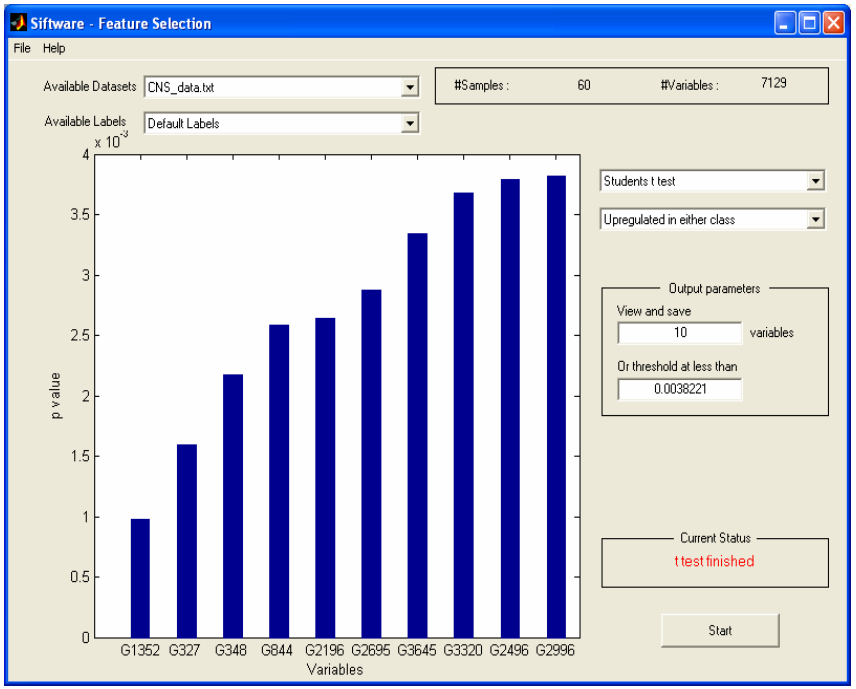




- Kasabov, N., *Modelling and profile discovery in Bioinformatics: Global, local and personalised approach,* Pattern Recognition Letters, Jan. 2007

Example: Finding significant genes of brain cancer survival based on gene expression data

(data from Pomeroy et al, Nature, 2000)



Kasabov, N., L.Benuskova, V.Jain, Integrating brain-gene ontology with evolving connectionist system for modelling and discovery, Neural Networks, accepted, 2007

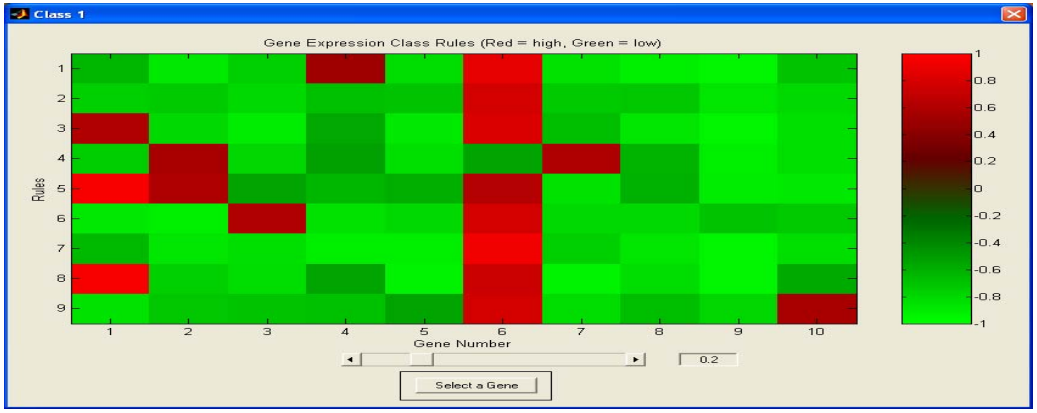
10 genes found to discriminate well the two classes (responding and not responding to a drug):

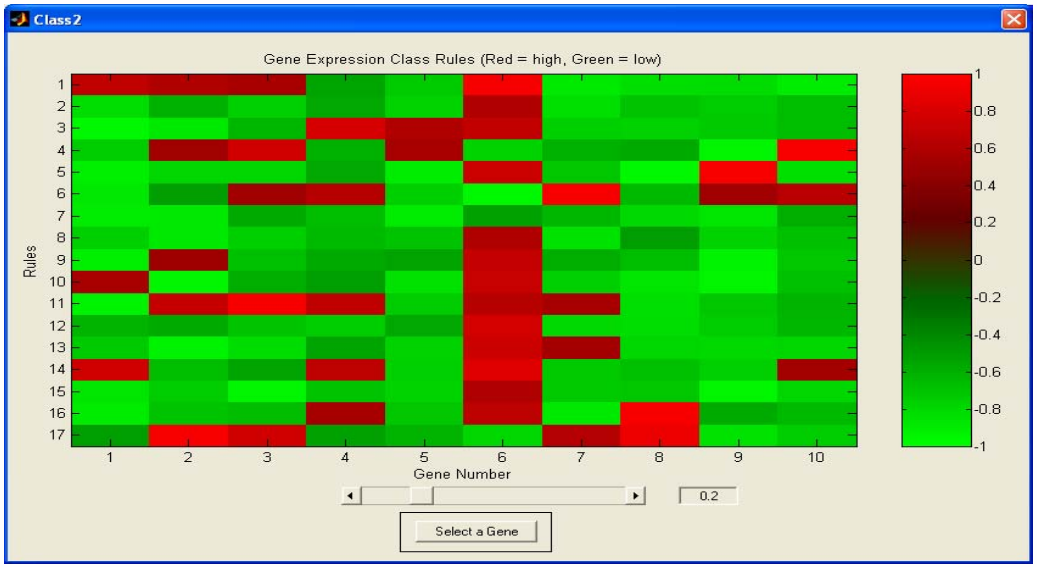
- G1 - G1352 = High mobility group protein (HMG-I(Y)) gene exons 1-8, L17131, high mobility group AT-hook 1, HMGA1
- G2 - G327 = D28124, NBL1 - neuroblastoma, suppression of tumorigenicity 1
- G3 - G348 = Probable Ubiquitin Carboxyl-terminal Hydrolase, D29956 UBPY (ubiquitin specific peptidase 8, USP 8)
- G4 – G844 = Dynein, Heavy Chain, Cytoplasmic, HG2417-HT2513
- G5 - G2196 = Polyposis Locus Protein 1, M73547, adenomatosis polyposis coli, APC
- G6 – G2695 = TAR (HIV-1) RNA binding protein 2, U08998, TARBP2
- G7 – G3645 = Prostaglandin transporter hPGT mRNA, U70867
- G8 – G3320 = Leukotriene C4 synthase (LTC4S) gene, U50136
- G9- G2496 = NTRK3 Neurotrophic tyrosine kinase, receptor, type 3 (TrkC), S76475 - (1 of 50 *markers of survival from Pomeroy et al. 2002*)
- G10 - G2996 = Gps2 (GPS2, G protein pathway suppressor 2) mRNA, U2896

Local and individual profiles of the two classes are derived using ECOS ->

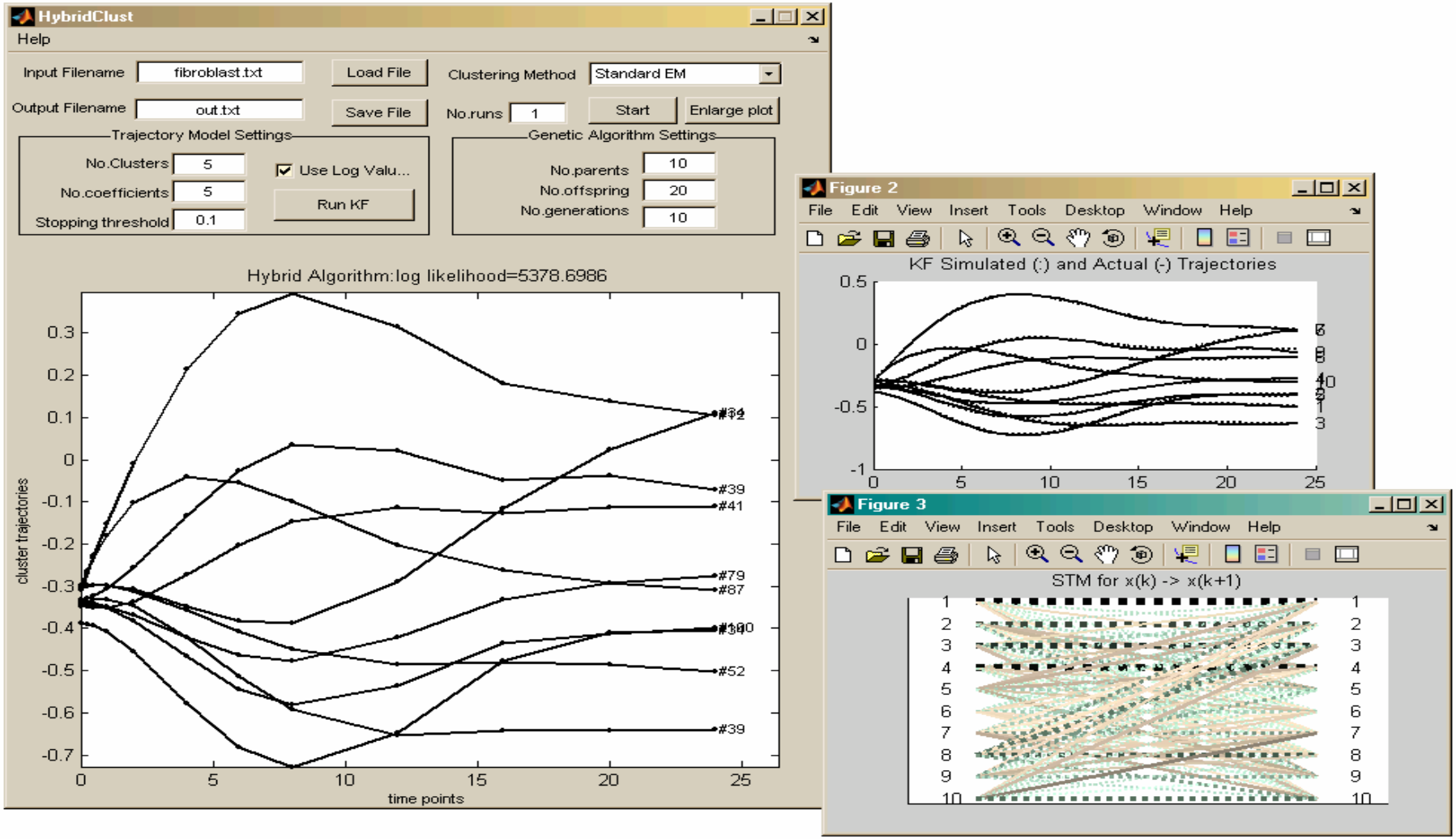
Example: Outcome profiling and prognosis of brain cancer based on gene expression data

(data from Pomeroy et al, Nature, 2000)





Gene regulatory network modelling
GeneNetXP (Chan, Jain et al, 2006)



Evolving fuzzy neural networks for GRN modeling

(Kasabov and Dimitrov, ICONIP, 2002)

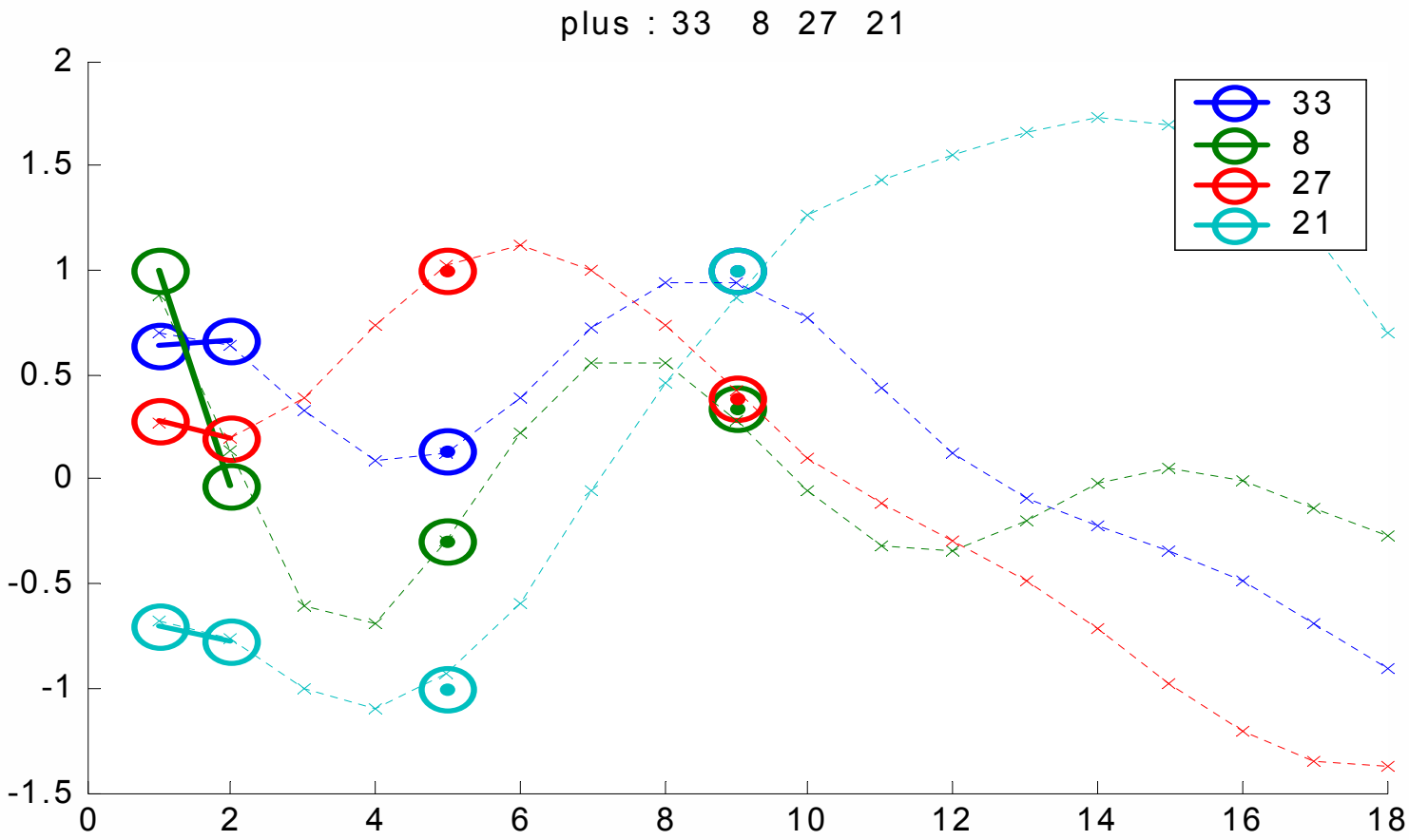
- On-line, incremental learning of a GRN
- Adding new inputs/outputs (new genes)
- The rule nodes capture clusters of input genes that are related to the output genes
- Rules can be extracted that explain the relationship between G(t) and G(t+dt),

e.g.: *IF g13(t) is High (0.87) and g23(t) is Low (0.9)*

THEN g87 (t+dt) is High (0.6) and g103(t+dt) is Low

- Playing with the threshold will give stronger or weaker patterns of relationship

Using a GRN model to predict the expression of genes in a future time

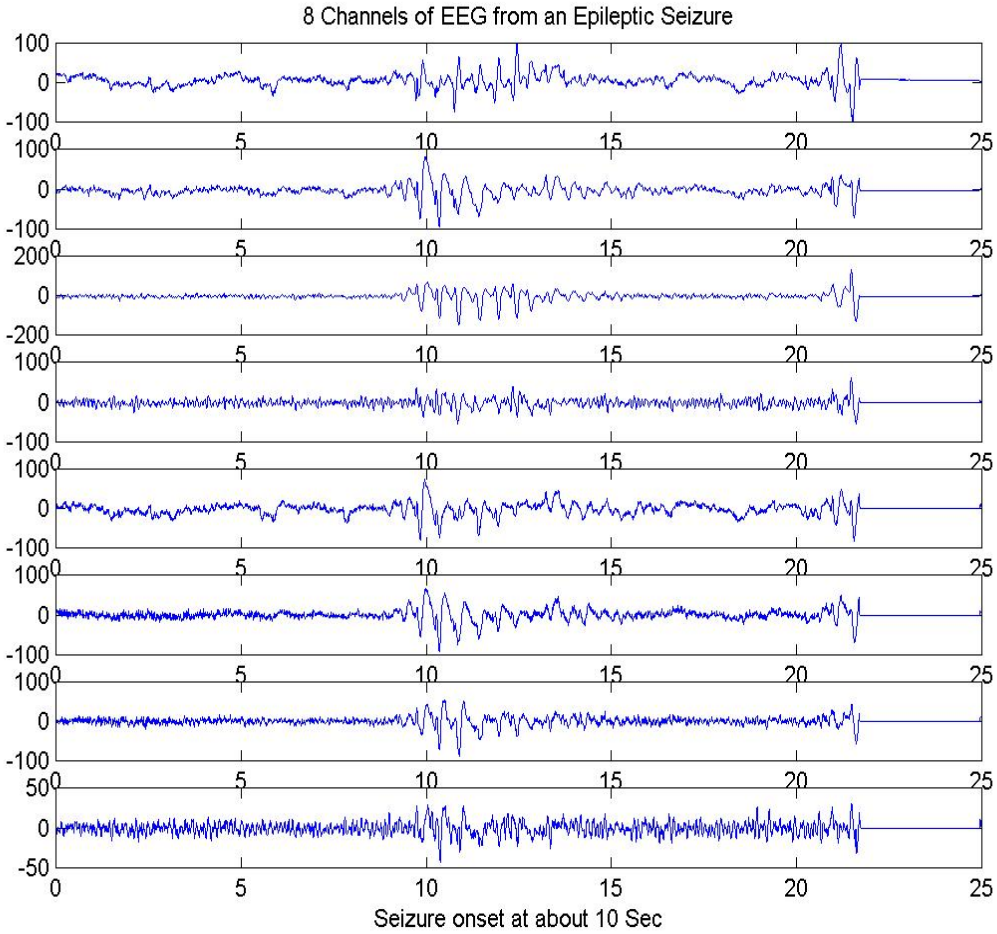


nkasabov@aut.ac.nz

9. Neuroinformatics and Neurogenetic Modelling

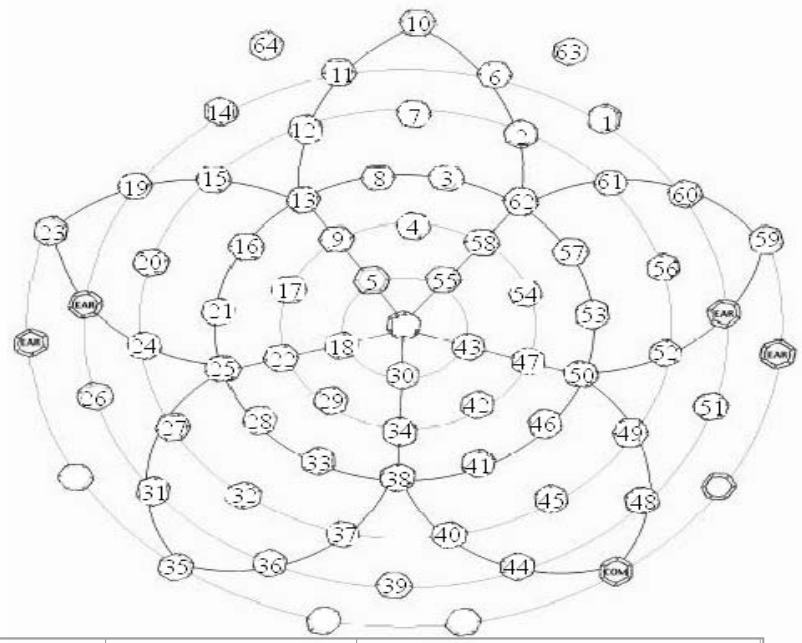
EEG data modelling

- Why ECOS for brain study'
- Modeling brain states of an individual or groups of individuals from EEG, fMR and other information
- Example: epileptic seizure a patient; 8 EEG channels data is shown
- Future directions:
  - Dynamic models for brain-computer interfac
  - Brain areas interaction networks
- Applications:
  - Neuroscience
  - Engineering



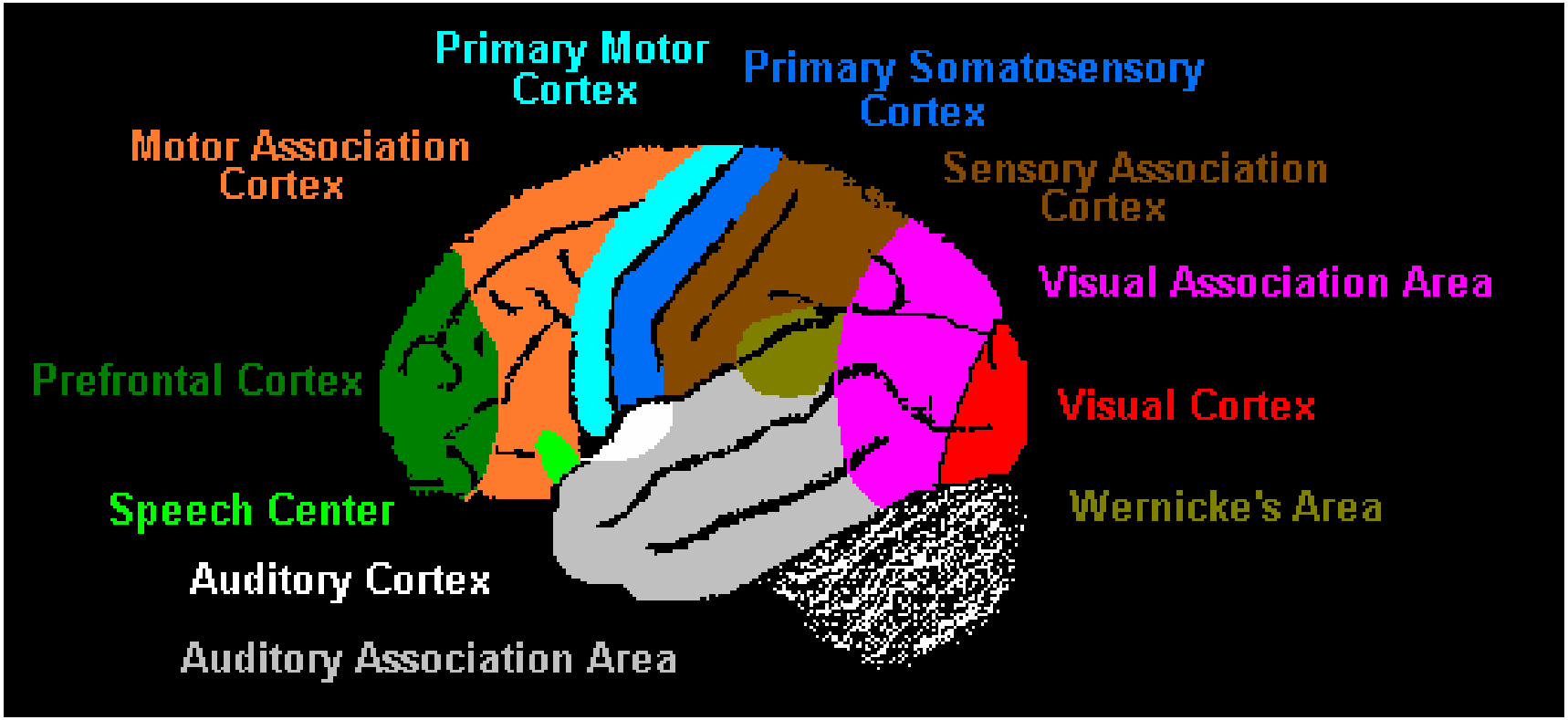
ECOS for modeling perception states, based on EEG data

- Standard EEG electrode systems
- In the experiment here, four classes of brain perception states are used with 37 single trials each of them including the following stimuli:
  - Class1 - Auditory Stimulus;
  - Class2 - Visual Stimulus;
  - Class3 - Mixed Auditory and visual stimuli;
  - Class 4 - No stimulus.
- The task is easy for a person but can a computer system do it in an automated mode?
- Brain-computer interfaces
- With van Leewen, Cihotsky et al, RIKEN, BSI, Tokyo

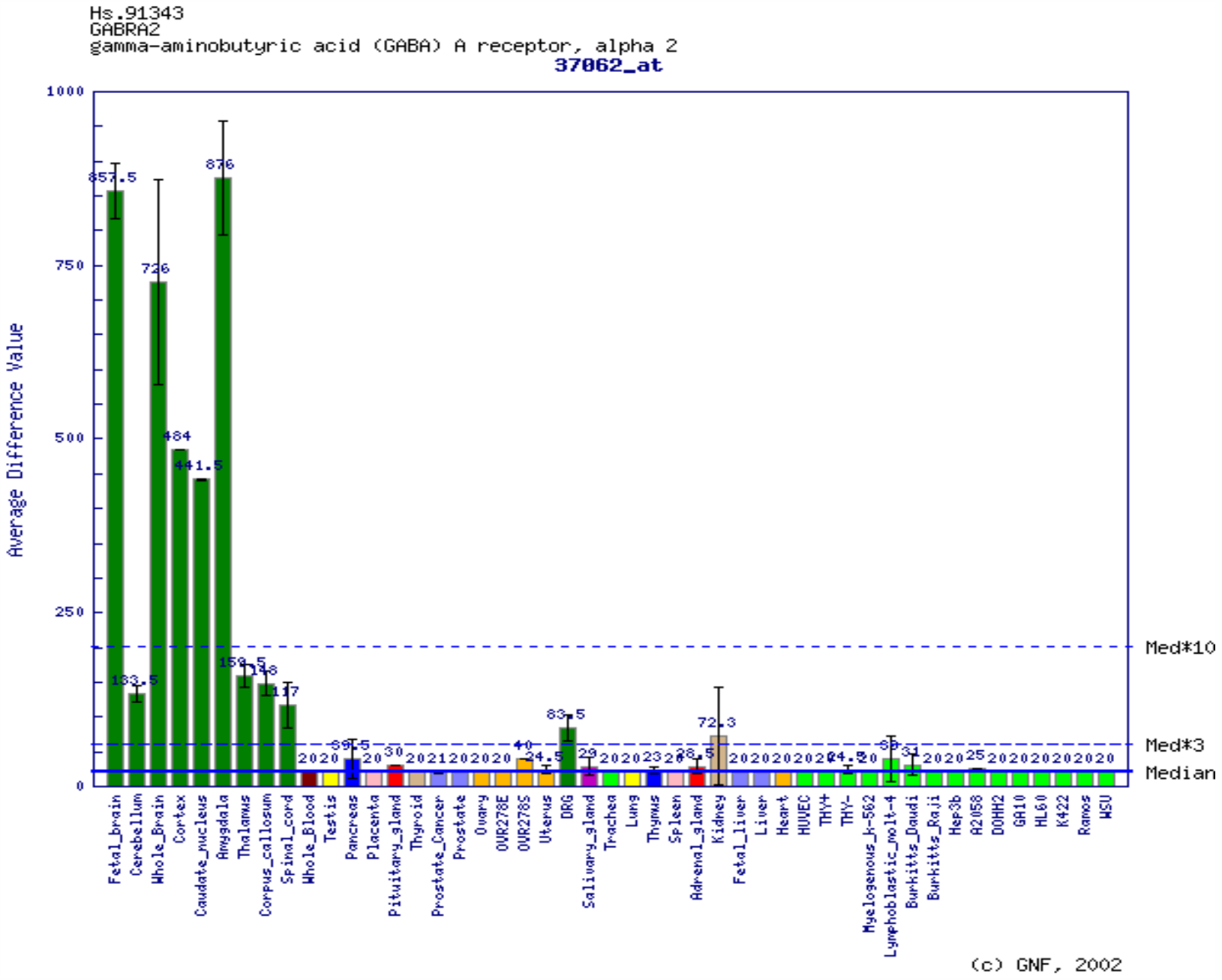


Stimulus	A	V	AV	No	Accuracy
A	81.2	1.3	0.1	0.2	98%
V	1.1	82.4	2.9	1.8	93.4%
AV	0.6	3.3	75	1.4	93.4%
No	0.4	1.5	1.3	80.5	96.2%

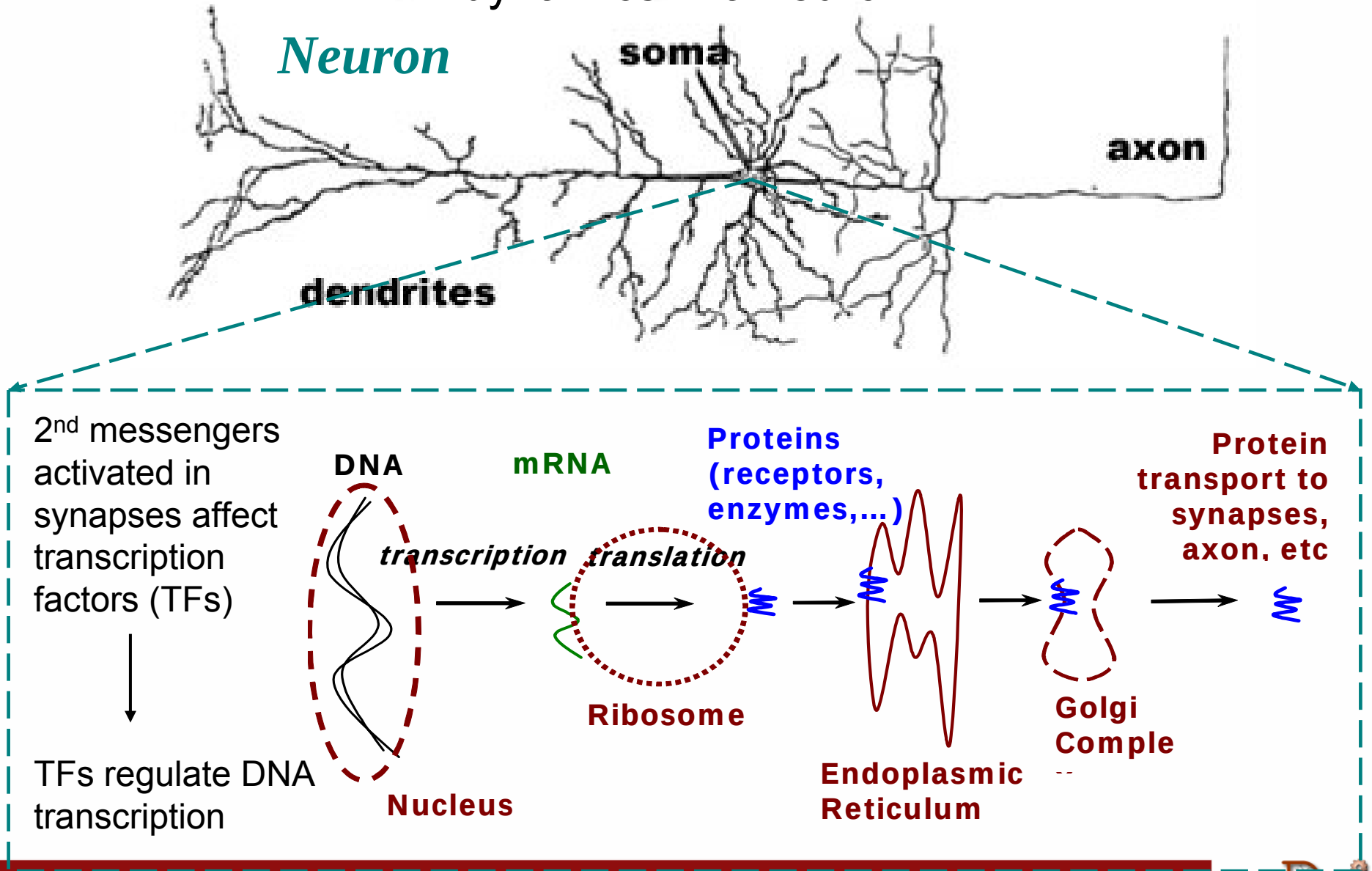
Neurogenetic Modelling.



Brain areas are specialized. Specialization is genetically determined, but a proper experience must occur early in life to gain that function.


Hs.91343
GABRA2
gamma-aminobutyric acid (GABA) A receptor, alpha 2
37062_at
Average Difference Value
1000
750
500
250
0
Med*10
Med*3
Median
Fetal_brain
Cerebellum
Whole_Brain
Cortex
Caudate_nucleus
Amygdala
Thalamus
Corpus_callosum
Spinal_cord
Whole_Blood
Testis
Pancreas
Placenta
Pituitary_gland
Thyroid
Prostate_Cancer
Prostate
Ovary
OVR278E
OVR278S
Uterus
DRG
Salivary_gland
Trachea
Lung
Thymus
Spleen
Adrenal_gland
Kidney
Fetal_liver
Liver
Heart
HUVEC
THY+
THY-
Myelogenous_K-562
Lymphoblastic_molt-4
Burkitts_Daudi
Burkitts_Raji
Hep3b
A2058
DOHH2
GA10
HL60
K422
Ramos
WSU
(c) GNF, 2002

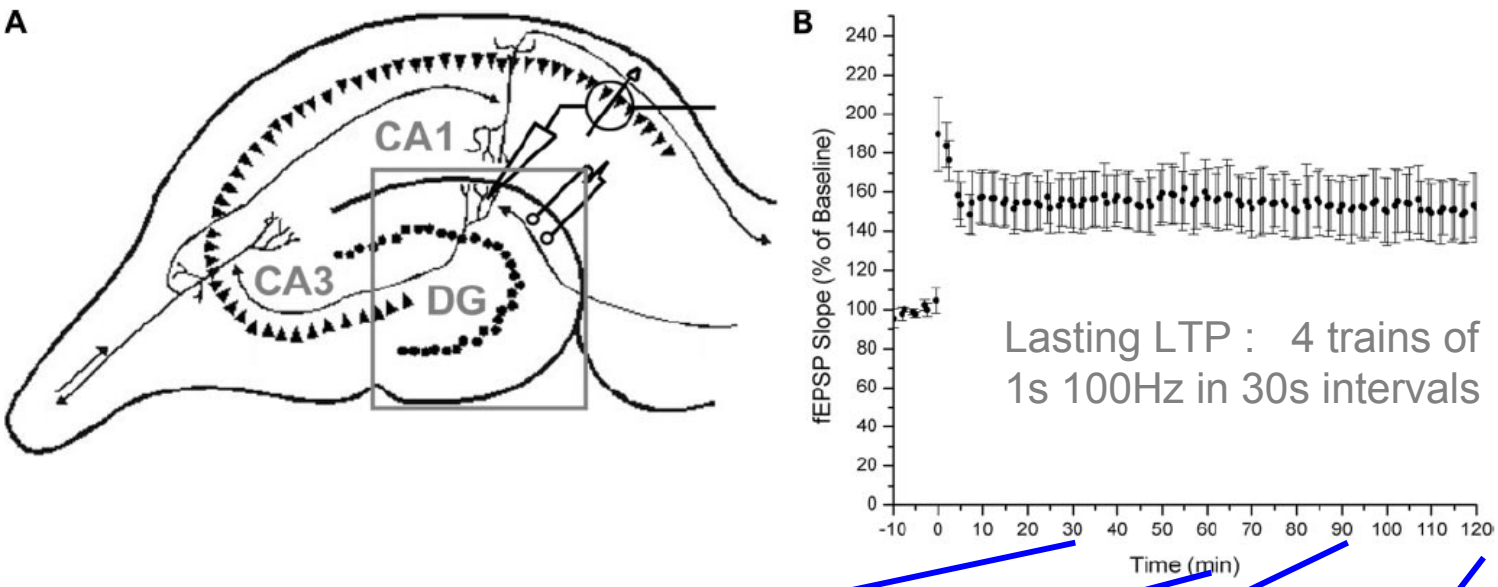
Interaction between fast neuronal dynamics with slow gene dynamics in a neuron



nkasabov@aut.ac.nz

Gene expression analysis related to learning and memory

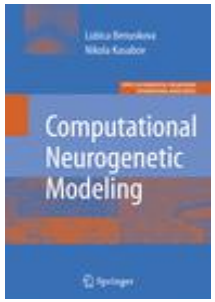
Park et al (2006) Molecular Network and Chromosomal Clustering of Genes Involved in Synaptic Plasticity in Hippocampus, J Biol Chem. 2006 Oct 6;281(40):30195-211



Time course DNA microarray analysis to determine the temporal genomic expression profiles of 12,000 genes

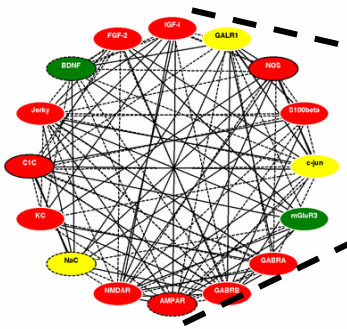
Computational Neurogenetic Modelling (CNGM)

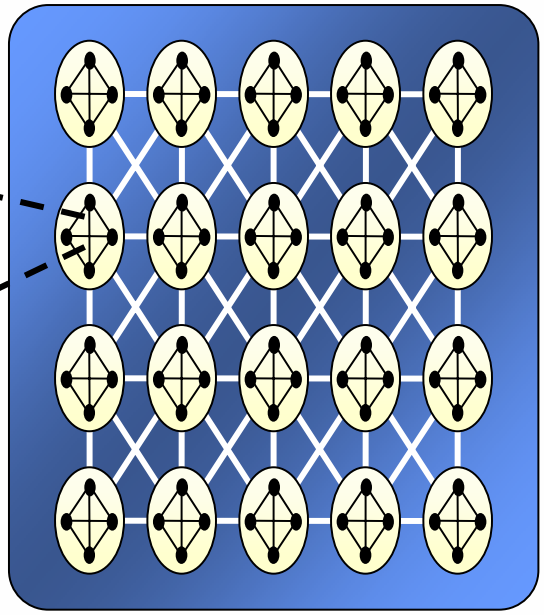
L.Benuskova and N.Kasabov, Springer, 2007

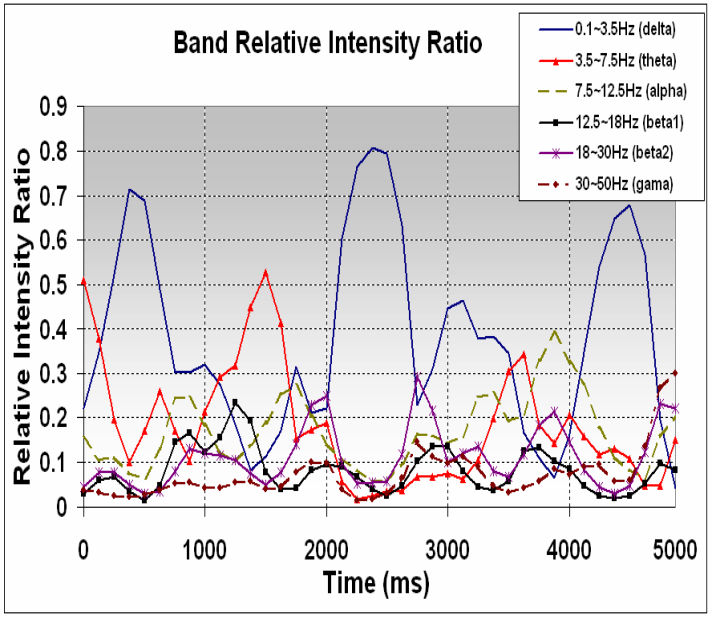


CNGM as an ANN

GRN

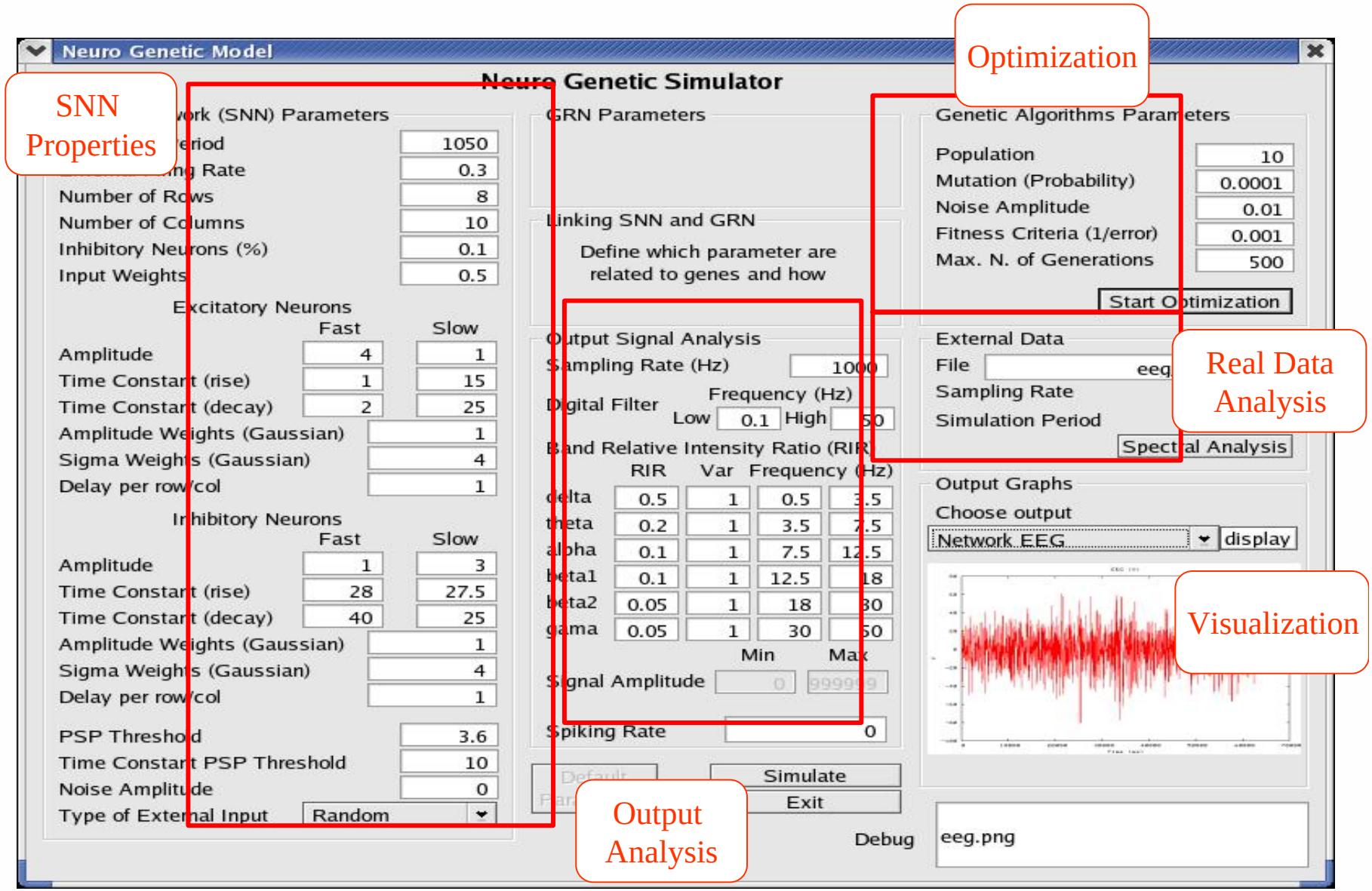






nkasabov@aut.ac.nz

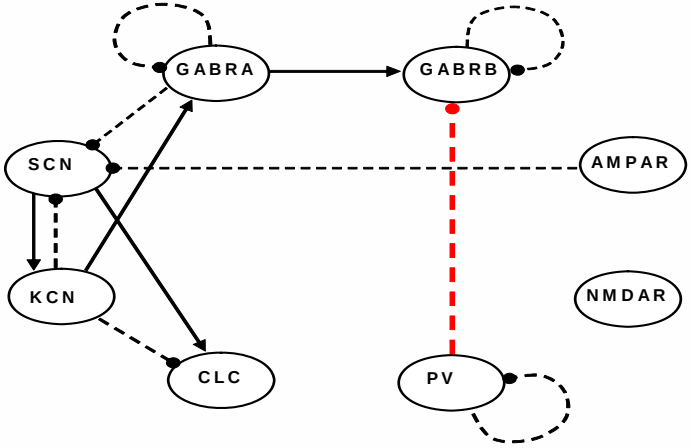
CNG Simulator (Available from KEDRI, www.kedri.info)



GRN derived from mouse epilepsy data (with A.Villa, U.Lausanne)

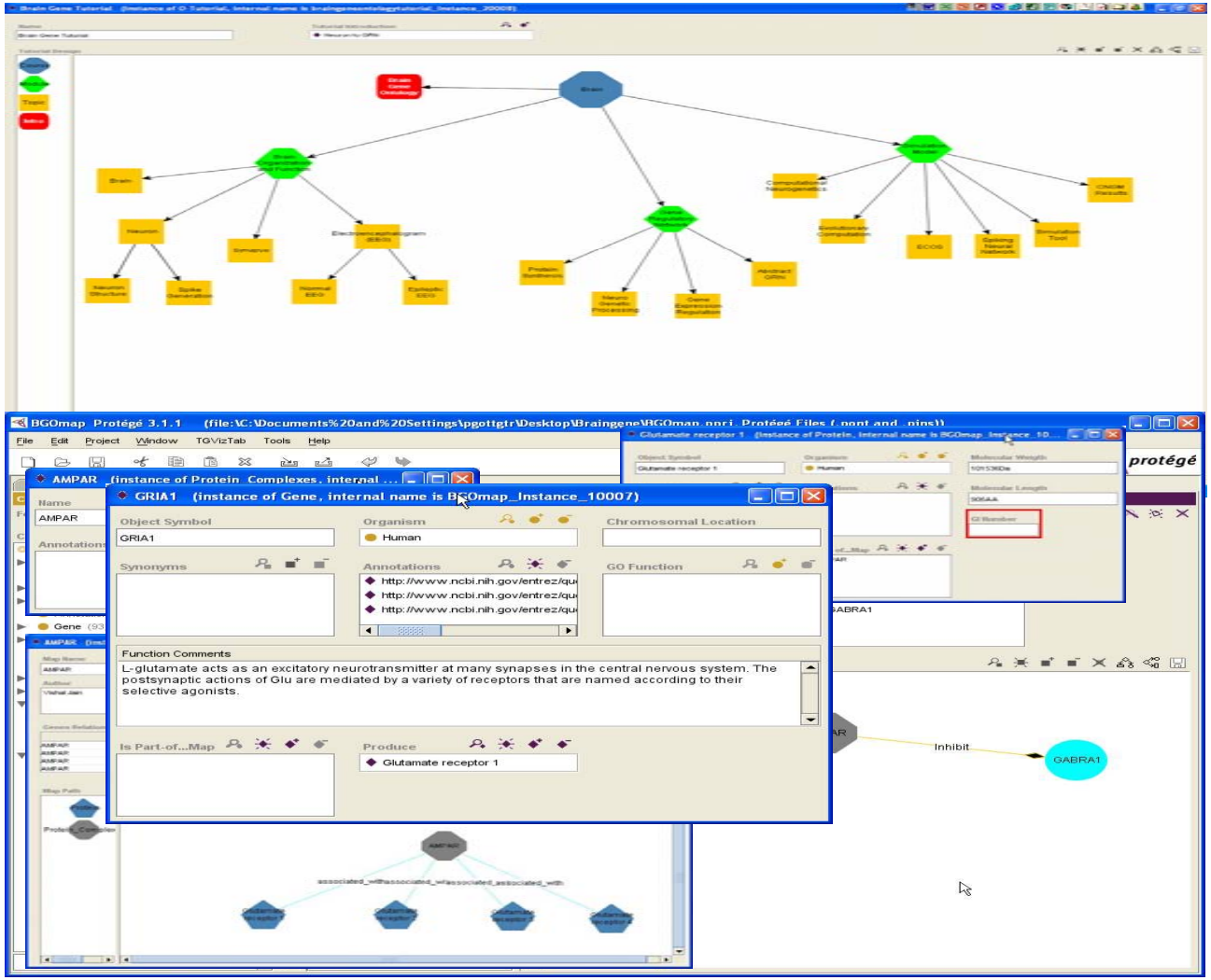
Table. Neuronal Parameters and Related Proteins

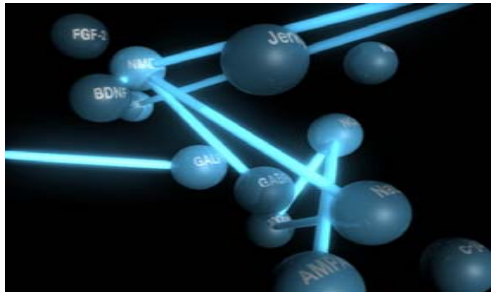
Neuronal parameter Amplitude and time constants of	Protein
Fast excitation PSP	AMPAR
Slow excitation PSP	NMDAR
Fast inhibition PSP	GABRA
Slow inhibition PSP	GABRB
Firing threshold	SCN, KCN, CLC
Late excitatory PSP through GABRA	PV

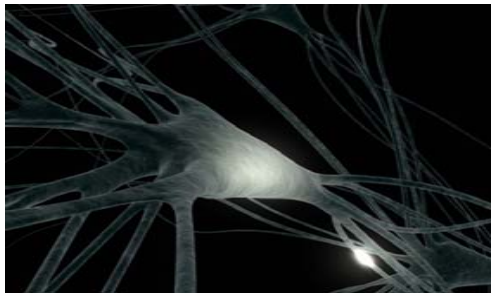


BGO: Brain-gene Ontology System

(IJCNN 2007, Neural Networks, to appear)

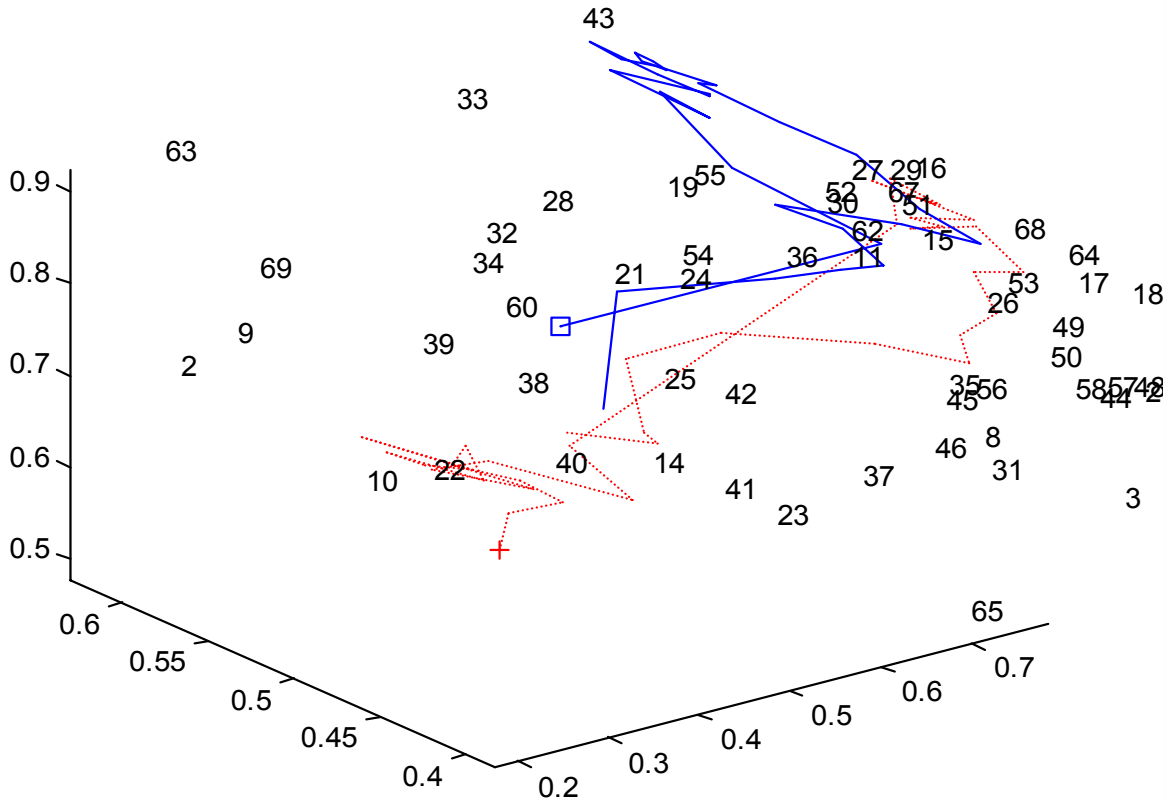






nkasabov@aut.ac.nz

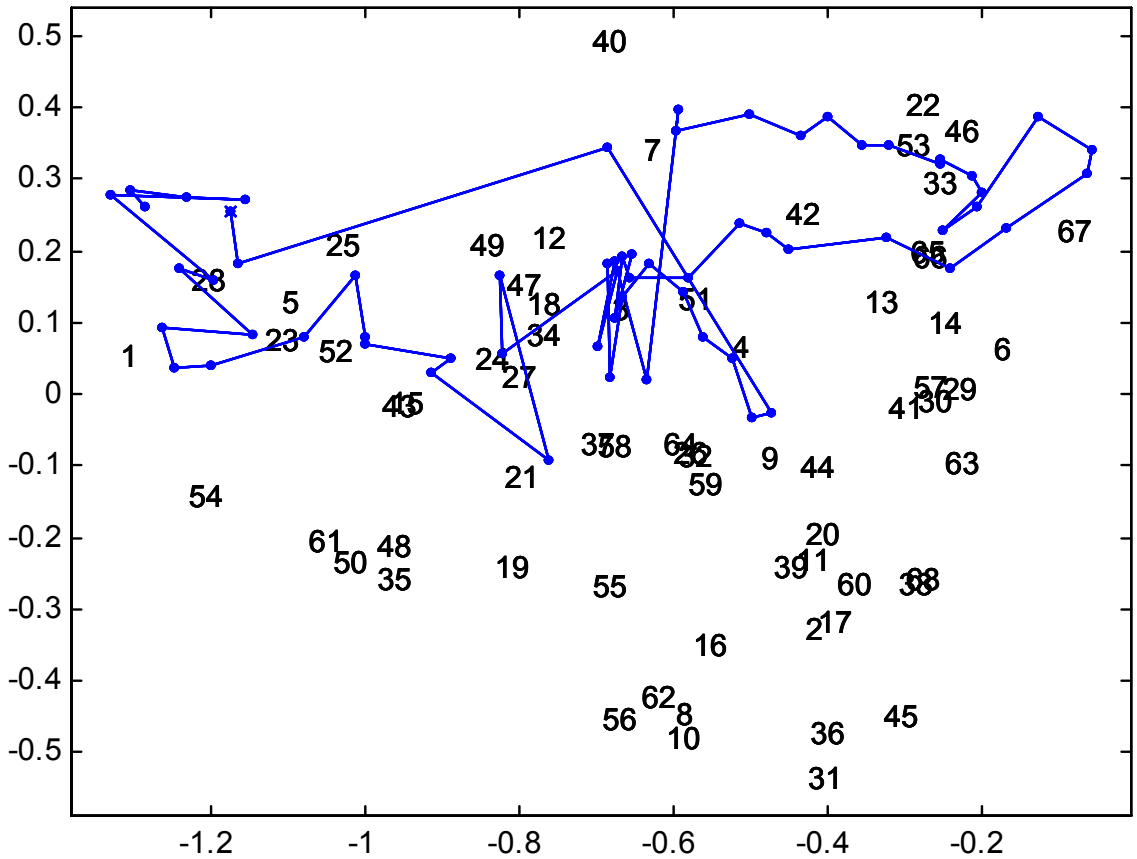
10. Modelling the emergence of acoustic segments in spoken languages



Trajectories of spoken words ‘sue’ and ‘nine’ by the same speaker presented in a 3D space MS1-MS7-logE, along with the 70 rules nodes of the evolved ECM model (Taylor et al, 2000).

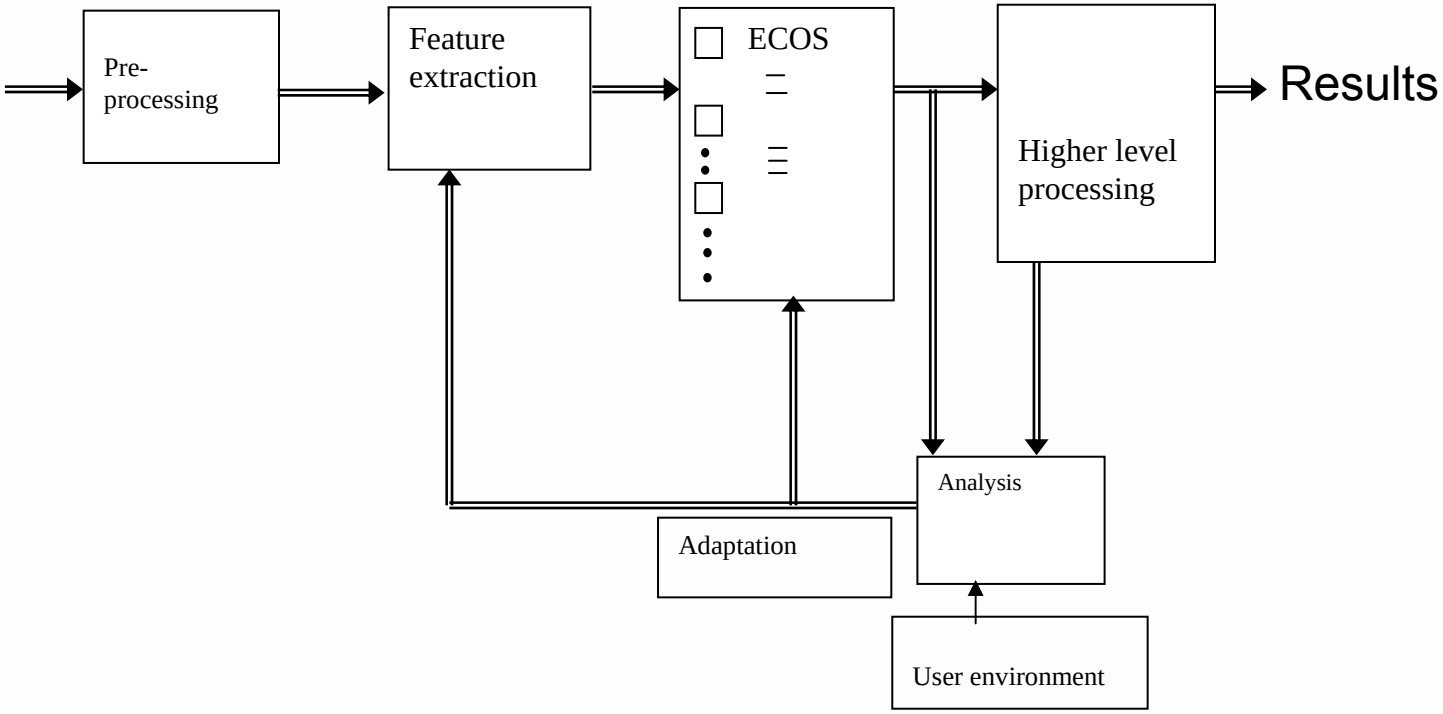
Evolving bilingual acoustic spaces

The projection of the spoken word “zoo” in English + Maori PCA space (see Laws,2001)



11. Adaptive speech recognition

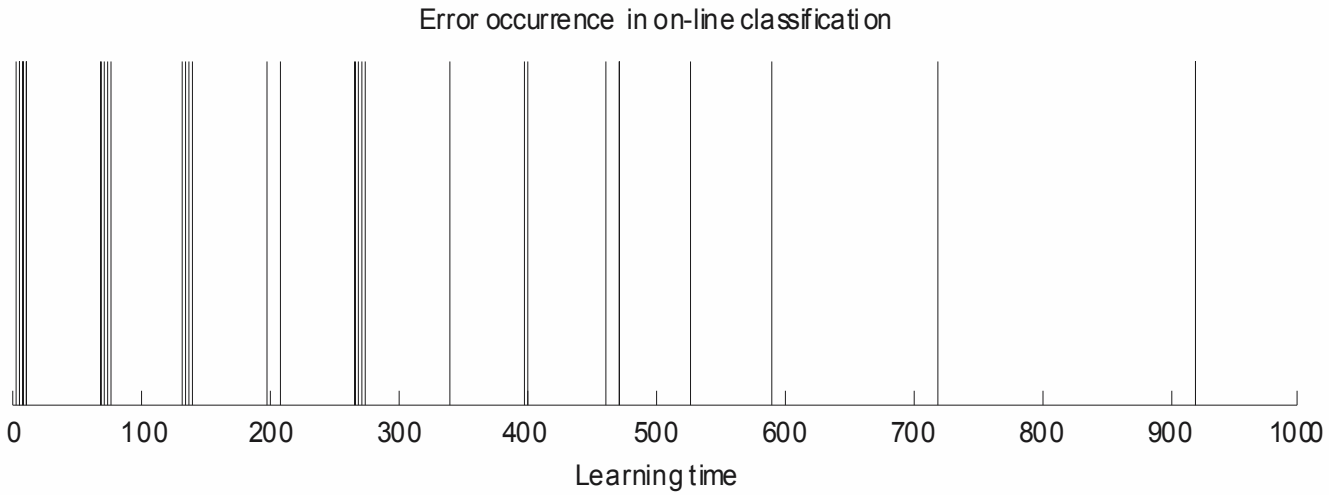
A Framework of Evolving Connectionist Systems for Adaptive Speech Recognition



A block diagram of an adaptive speech recognition system framework that utilises ECOS in the recognition part

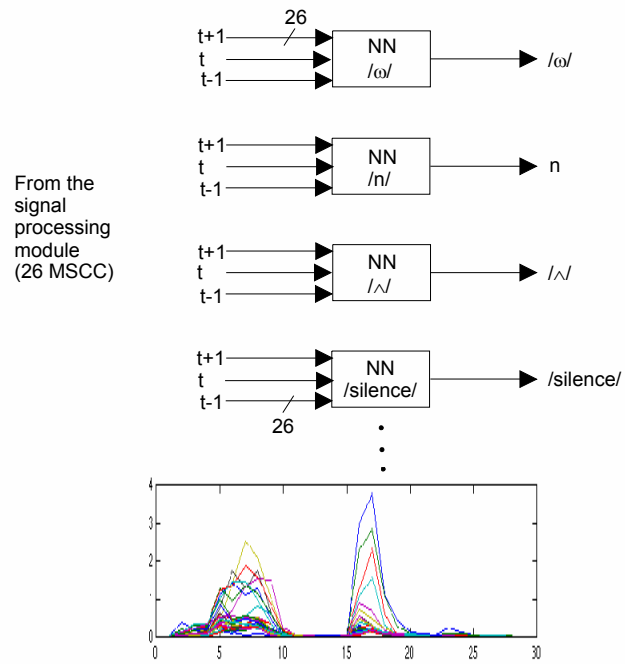
ESOM for On-line Adaptive Phoneme Classification

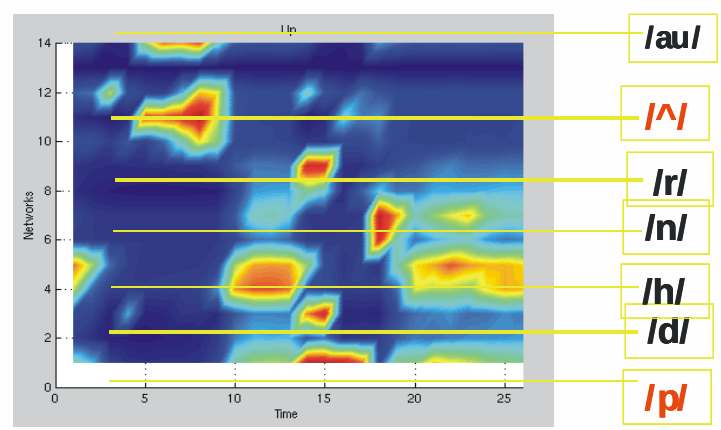
- ESOM was used for the classification of phoneme data
- advantage of ESOMs as classifiers is that they can be trained (evolved) in a life-long mode
- Longer system is training, the lower the error rate
- *Error rate of an ESOM system trained in an online learning mode (Fig. 11.7)*



Multimodel phoneme recognition

- One NN module is trained on a single phoneme data
- Single phoneme NN can be further adapted to different accents and pronunciations without necessarily re-training the whole system
- Phoneme modules miss-activation problem can be overcome through analysis of the sequence of the recognised phonemes and forming the recognized word through a matching process using a dictionary of words.
- To improve the recognition rate, the wrongly activated phoneme NN modules can be further trained not to react positively on the problematic for them phoneme sounds.
- New NN modules can be created to capture new phonemes
- Each of the phoneme NN module can be further adapted to a new accent, e.g. Australian English





On-line digit recognition – English digits

- Recognition of speaker independent pronunciations of English digits
- 17 speakers (12 males and 5 females) are used for training,
- 17 other speakers (12 males and 5 females) are used for testing
- Comparison of two NN models – ECOS and Linear Vector Quantization (LVQ)
- Training is performed on clean data and testing – on noisy data
- Car noise and office noise are added to the clean speech for testing
- With IRST Trento, Italy, Edmondo Trentin

12. Adaptive image processing

The visual system

- *The visual system is composed of eyes, optic nerves, many specialised areas of the cortex (the ape for example has more than 30).*

- *The image on the retina is transmitted via the optic nerves to the first visual cortex (V1), which is situated in the posterior lobe of the brain. There the information is divided into two main streams, the "what" tract and the "where" tract.*

- *The ventrical tract ("what") separates targets (objects and things) in the field of vision and identifies them. The tract traverses from the occipital lobe to the temporal lobe (behind the ears).*
- *The dorsal tract ("where") is specialised in following the location and position of the objects in the surrounding space. The dorsal tract traverses from the back of the head to the top of the head.*

- *How and where is the information from the two tracts united to form one complete perception, is not completely known.*

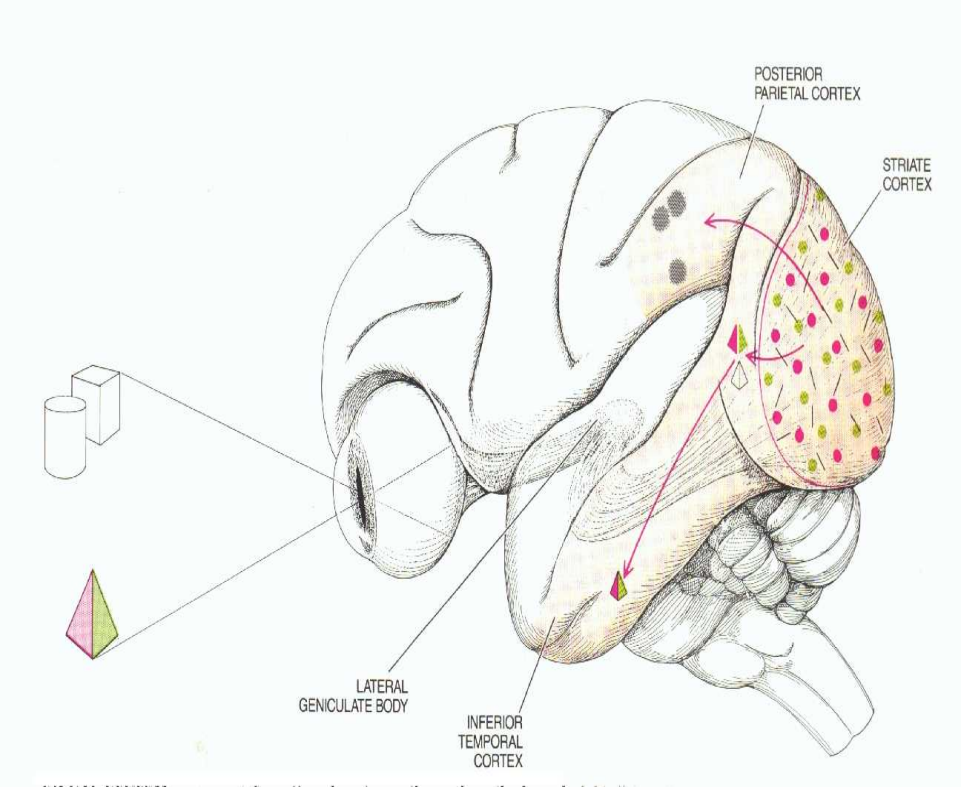
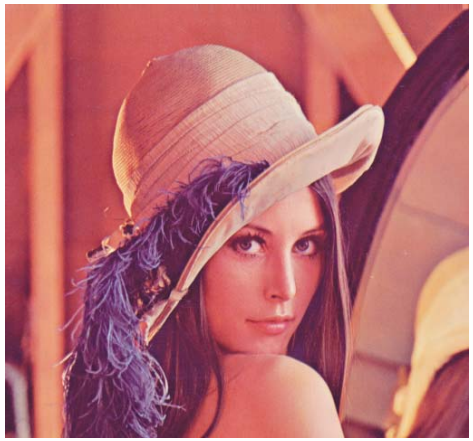


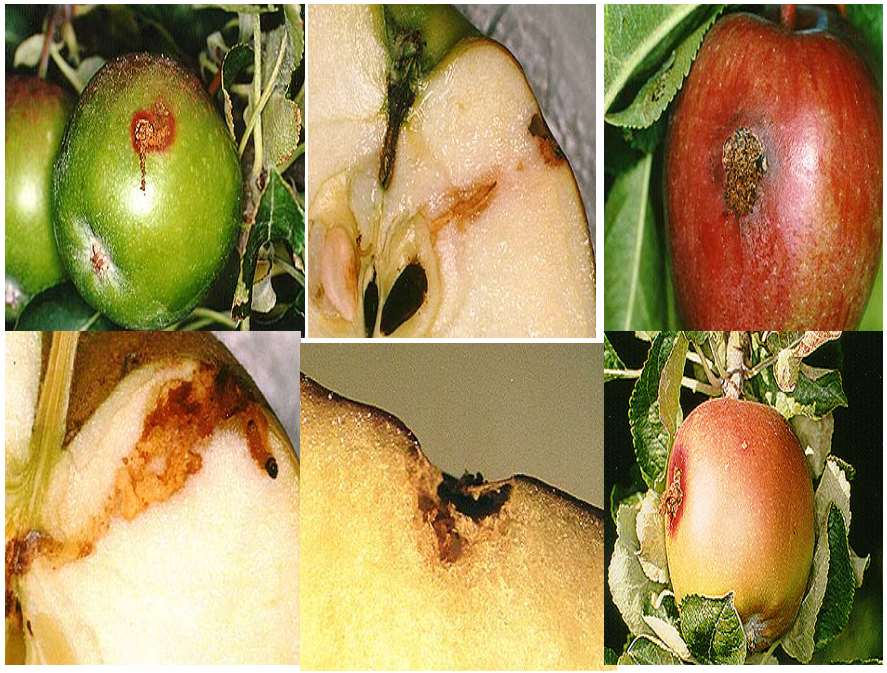
Image colour quantisation through ESOM clustering

- An original image of a large number of colours (e.g. 16 777 216 = 256*256*256, as each of the R, G and B colours has 256 levels) → see Lena on top →
- The task is: The image to be represented by a smaller number of colours (e.g. 256) but retaining the quality of the image → see Lena, quantised at the bottom
- Clustering all colours of all pixels in the RGB 3D space
- Replacing each colour in each pixel from the original image with its closest colour cluster centre (prototype)
- Quantisation error
- Other examples are presented in the book



Practical Example: Fruit insect identification based on images from damaged apples

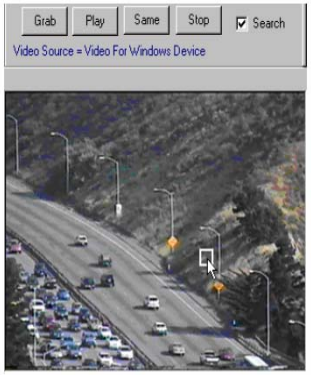
- 5 apple pests
- 67 images of damaged apples used for training, labelled with the name of the pest (1 to 5,e.g. codling moth)
- 23 images used for testing (labelled)
- Features: wavelengths
- Classification method: EFuNN
- Overall test result: 61%



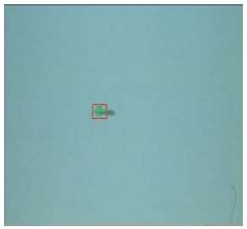
Adaptive image recognition and object tracking

Target tracking

- Learn one or several target
- Detect similar pattern in subsequent frame
- Re-inforced learning
- Smart scanning

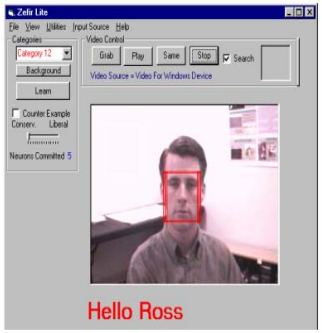


Automobile traffic control, aerial surveillance, human tracking, cell motion analysis



Biometrics

- Person authentication
- Facial expression recognition

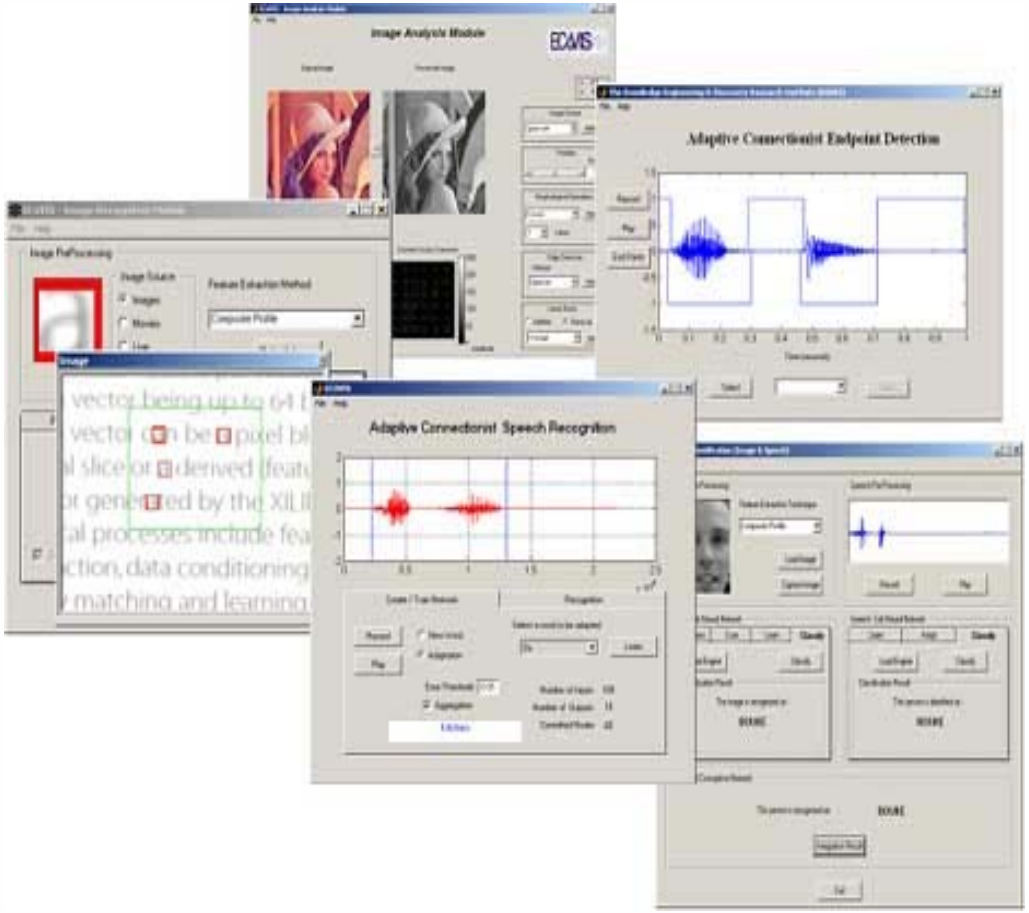


ZISC Manual, Silicon Recognition Ltd,

13. Adaptive Multimodal Information Processing

Adaptive audio-visual pattern recognition

- Adaptive speech recognition, image and video data processing
- ECOVIS prototype system
- Multimodal (face-, finger print-, iris-, and voice) person verification system
- Future research: Other modality recognition models and systems such as: odour-, DNA-, waves- , personal magnetic fields.



ECOS for multi-modal person identification from speech and image data streams (video data)

- The task is to identify persons from video information
- Integrating voice and face recognition
- Segmenting the video information
- Extracting dynamic image features that capture how the images change over time in the video data stream
- A NN is trained on **synchronised** 105 image features and 78 speech features
- The recognition, when integrated input data is used, is better than when either speech or image is used separately.
- Collaboration with the U.of Maastricht, The NL

Multiple modality systems are better than single modality systems

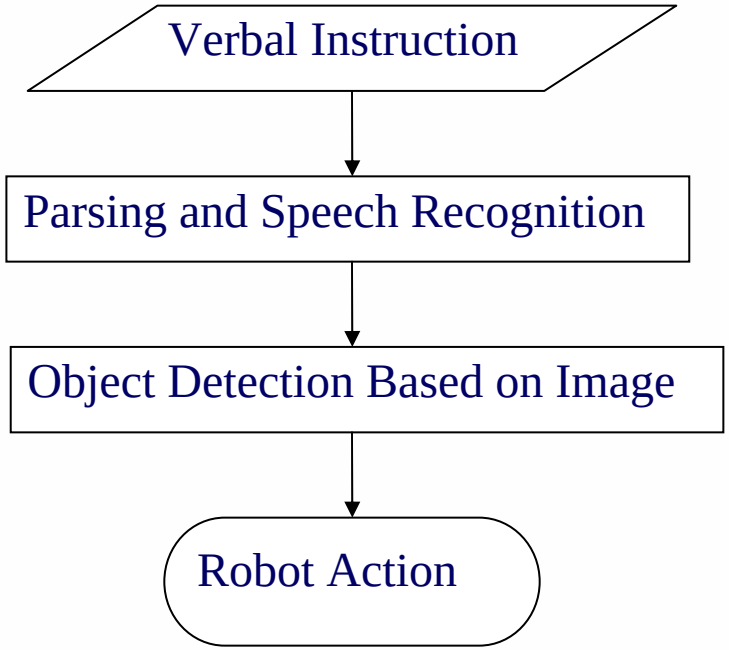
- The test (out of sample) results (the % of correct classification rate) of the classification of the 40 test frames by subsystems in a multimodal system (SS) for each of the 4 persons (P) from the CNN news program:

	P1	P2	P3	P4	%
Auditory SS only	7	5	3	9	60
Visual SS only	4	6	5	8	57.5
Early integration	7	8	3	8	67.5
Higher level integration	7	8	4	9	70

14. Adaptive Robotics and Decision Support Systems

ECOS for adaptive multimodal signal processing, speech and image

A simple case study: ECOS-based, adaptive, voice controlled object recognition system



	Speech Utterance
1	“Pen”
2	“Rubber”
3	“Cup”
4	“Orange”
5	“Circle”
6	“Ellipse”
7	“Rectangle”

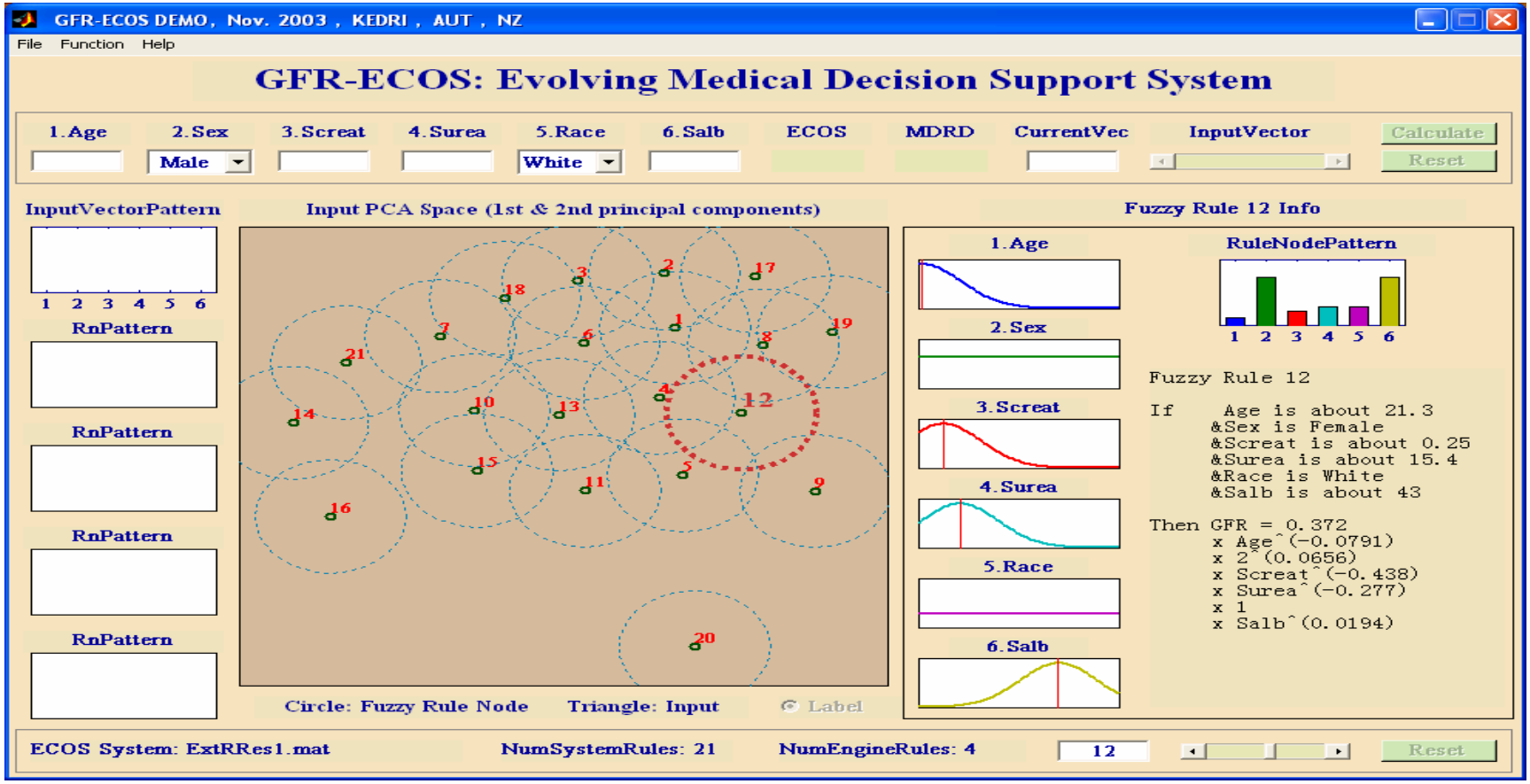
Robocup

1 gold and 2 silver medals for the Singapore Polytechnic at the World Robo-cup in Seoul, 2004, using

ECOS; Bronze medal in Japan, 2005 (Loulin)

Local, adaptive renal function evaluation system based on DENFIS:

(Marshal, Song, Ma, McDonell and Kasabov, Kidney International, May 2005)



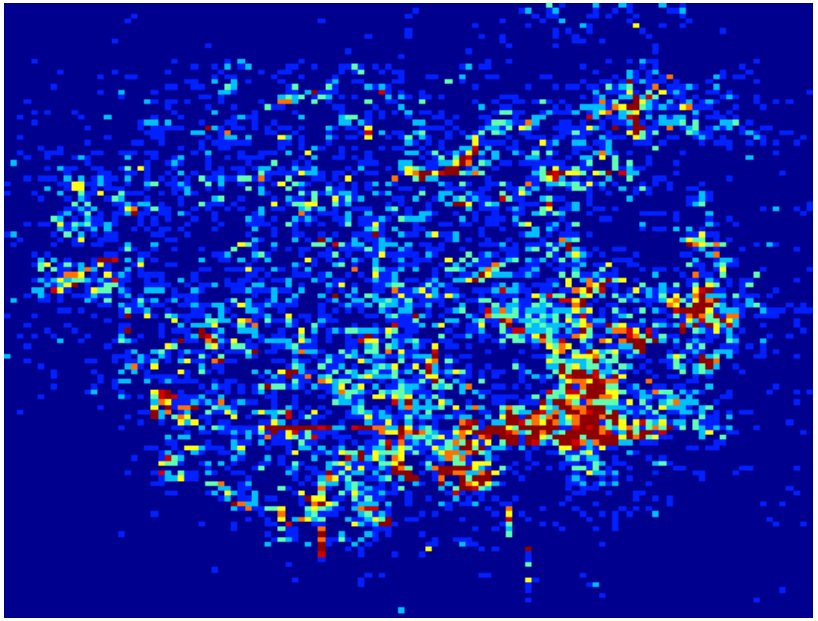
- **New method:** Song, Q. , N. Kasabov, T. Ma, M. Marshall, *Integrating regression formulas and kernel functions into locally adaptive knowledge-based neural networks: a case study on renal function evaluation*, Artificial Intelligence in Medicine, 2006, Volume 36, pp 235-244

Other On-line Decision Support Systems

- Mobile Calls Traffic Prediction and Optimization

- Farm milk volume prediction

- Radio-astronomy:
  - Extraterrestrial signal analysis
  - New planet discovery



15. Future directions: Quantum inspired CI

- **Quantum principles: superposition; entangelment, interference, parallelism**

- **QI methods for EIS:**
  - QI clustering
  - Quantum neuron with a recurrent connection (Li et al, 2006): the output and the input variables are represented as particle waves
  - QI neural networks (Ezov and Ventura, 2000)
  - QI associative memories (e.g. Ventura and Martinez, $m=O(2^n)$ patterns stored in 2n+1 neurons , while in a Hopfield NN it is $m<0.5n$ patterns in n neurons
  - QI fuzzy systems
  - QI GA
  - QI swarm intelligence
- **Applications:**
  - Specific algorithms with polynomial time complexity for NP-complete problems (e.g. factorising large numbers, Shor, 1997)
  - Search algorithms (Grover, 1996), $O(N^{1/2})$ vs O(N) complexity)
  - Memorizing large number of patterns

- *N.Kasabov, Brain-, Gene-, and Quantum Inspired Computational Intelligence: Challenges and Opportunities, in: W.Duch and J.Manzduk (eds) Challenges in Computational Intelligence, Springer, 2007*

Example: Quantum inspired evolutionary algorithms (QEA)

- The representation of individuals is usually done in the form of bit-strings, real-valued vectors, symbols etc. QEA uses a q-bit representation based on the concept of q-bits in Quantum Computing. Each q-bit is defined as a pair of numbers (*α, β*). A Q-bit individual as a string of *m* q-bits is represented as:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix}$$

for i = 1, 2, …, m:

$$|\alpha_i|^2 + |\beta_i|^2 = 1$$

- Evolutionary computing with Q-bit representation has a better characteristic of population diversity than other representations, since it can represent linear superposition of states probabilistically .

- Here, only one Q-bit individual with m q-bits is enough to represent 2^m states whereas in binary representation, 2^m individuals will be required for the same.
- The Q-bit representation leads to quantum parallelism in the system as it is able to evaluate the function on a superposition of possible inputs. The output obtained is also in the form of superposition which needs to be collapsed to get the actual solution.

- In QEA, the population of Q-bit individuals at time *t* can be represented as:

$$Q(t) = \{q_1^t, q_2^t, \ldots, q_n^t\}$$

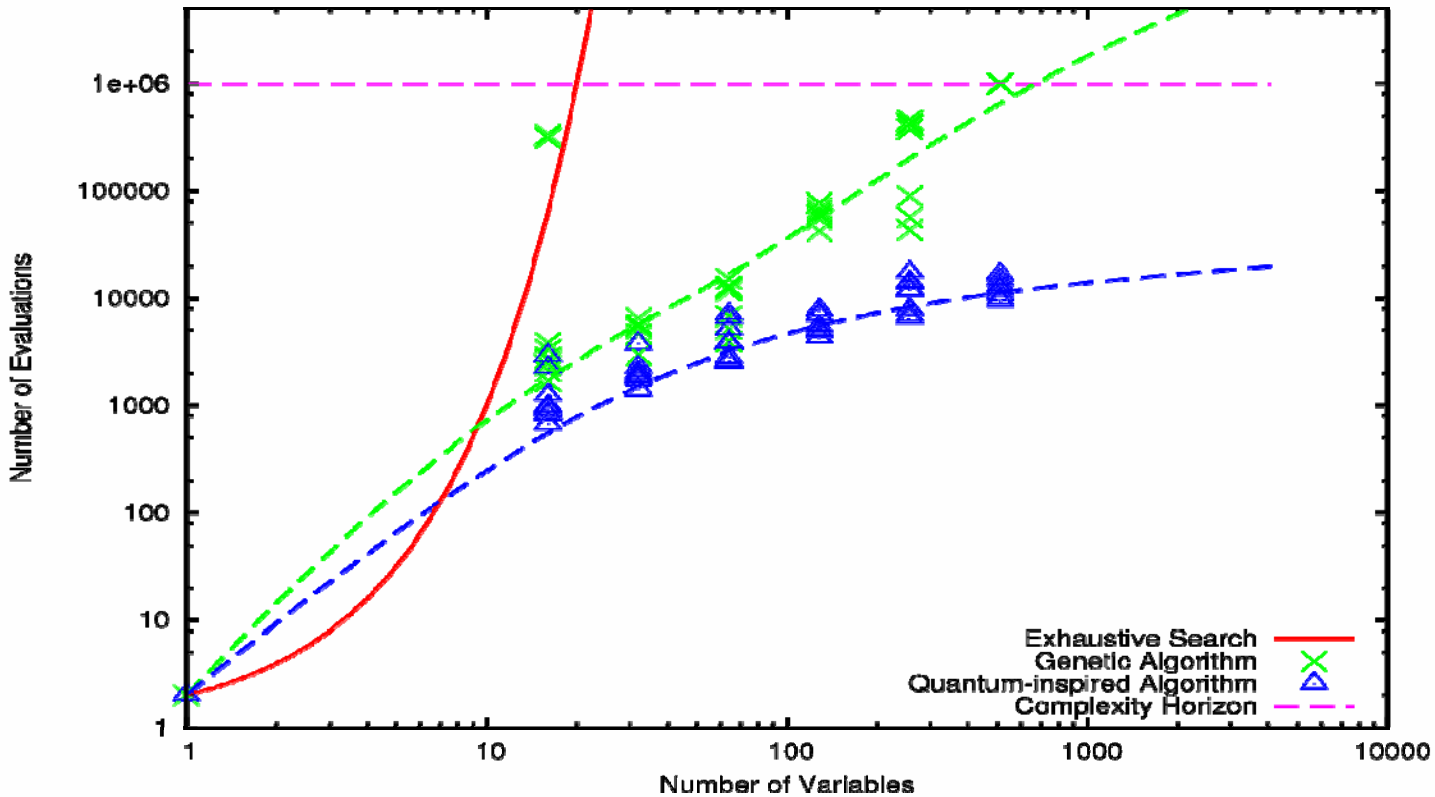
where n is the size of the population.

- The rotation gate, used as Q-gate is represented as:

$$U(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Flexible Quantum Inspired Evolutionary Algorithm

(M Defoin-Platel, S.Shliebs, Kasabov, Proc. CEC, 2007)



The KEDRI quantum inspired evolutionary algorithm performs exponentially faster and more accurately than the classical algorithms when evaluating combinations of variables for a modelling task

KEDRI: The Knowledge Engineering and Discovery Research Institute

- Established June 2002
- Funded by AUT, NERF (FRST), NZ industry
- External funds approx NZ$3.8 mln.
- 6 senior research fellows and post-docs
- 20 PhD and Masters students;
- 25 associated researchers
- Both fundamental and applied research (theory + practice)
- 150 refereed publications
- 2 PCT patents
- Multicultural environment (9 ethnic origins)
- Strong national and international collaboration