Advanced AI for Longevity Genomics

Dr. Ben Goertzel

OpenCog Foundation

Hong Kong Poly U

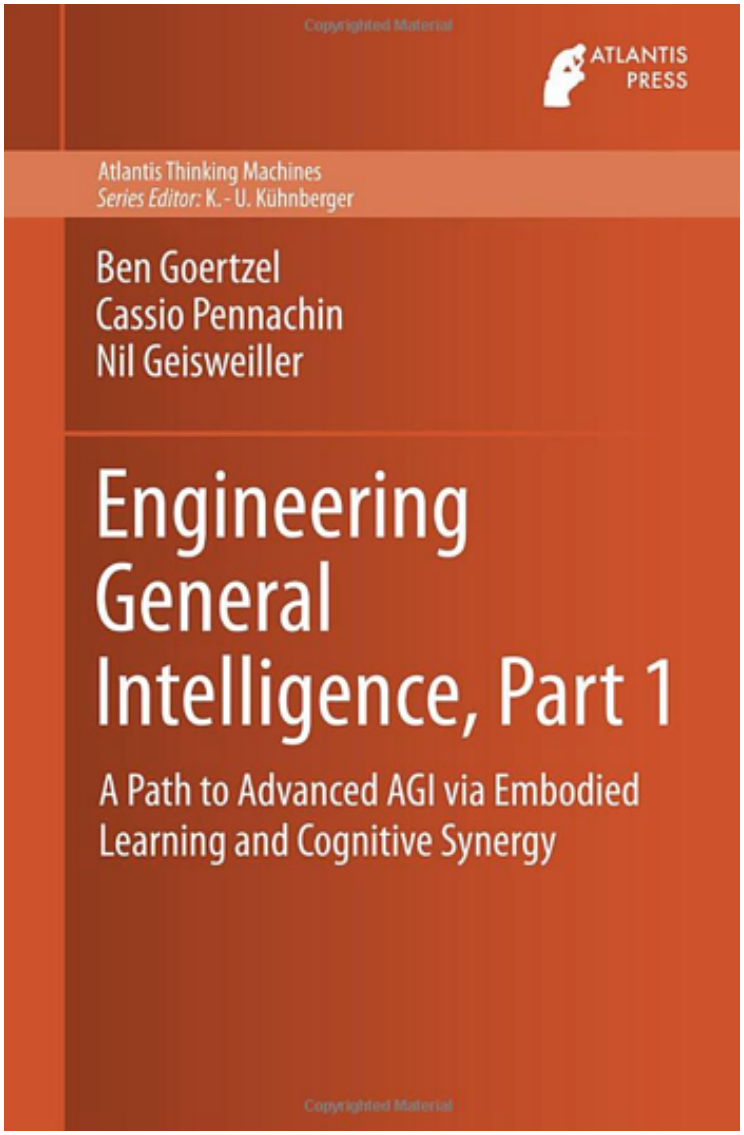
Hanson Robotics

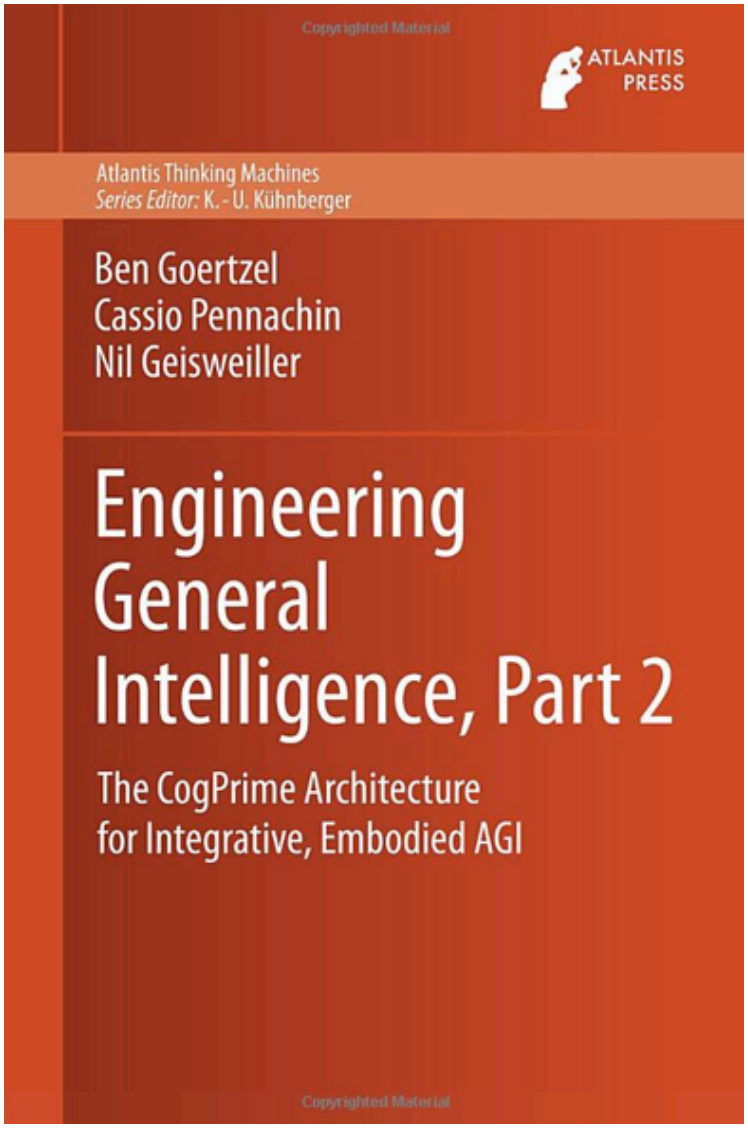
Aidyia Limited

Stevia First

OpenCog

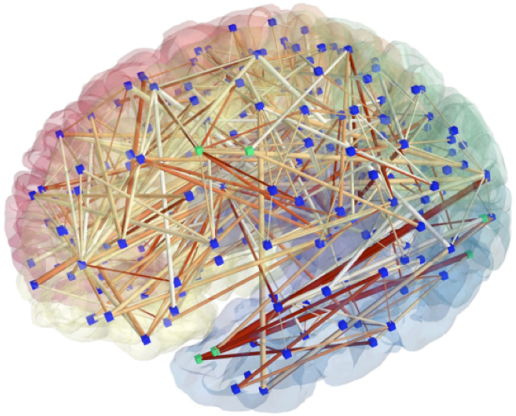
- Open-source AI project aimed at Artificial General Intelligence
- Integrated system aimed at controlling autonomous, generally intelligent agents
- Components of OpenCog currently in use for various practical applications…
- *… such as analyzing genomics data*

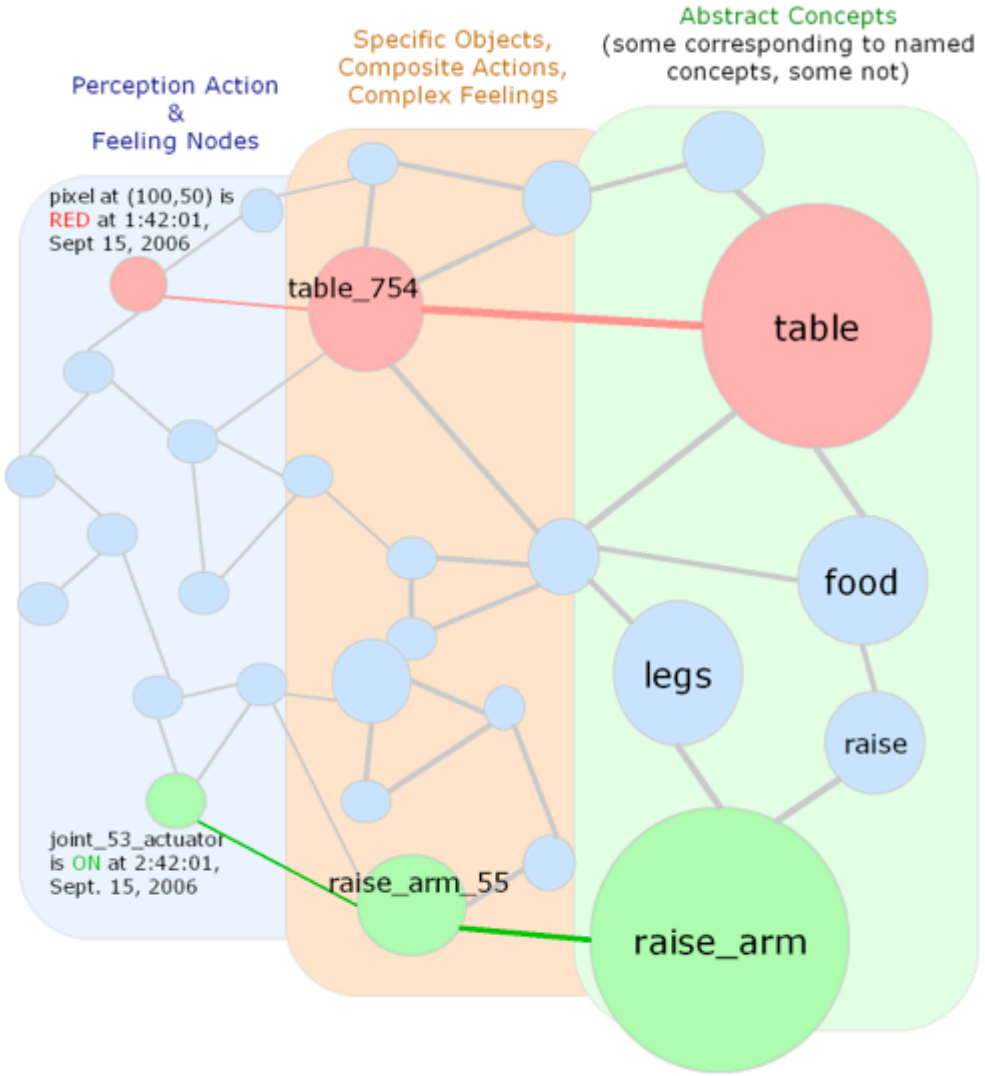

Copyrighted Material
ATLANTIS PRESS
Atlantis Thinking Machines
Series Editor: K.-U. Kühnberger
Ben Goertzel
Cassio Pennachin
Nil Geisweiller
Engineering General Intelligence, Part 1
A Path to Advanced AGI via Embodied Learning and Cognitive Synergy
Copyrighted Material

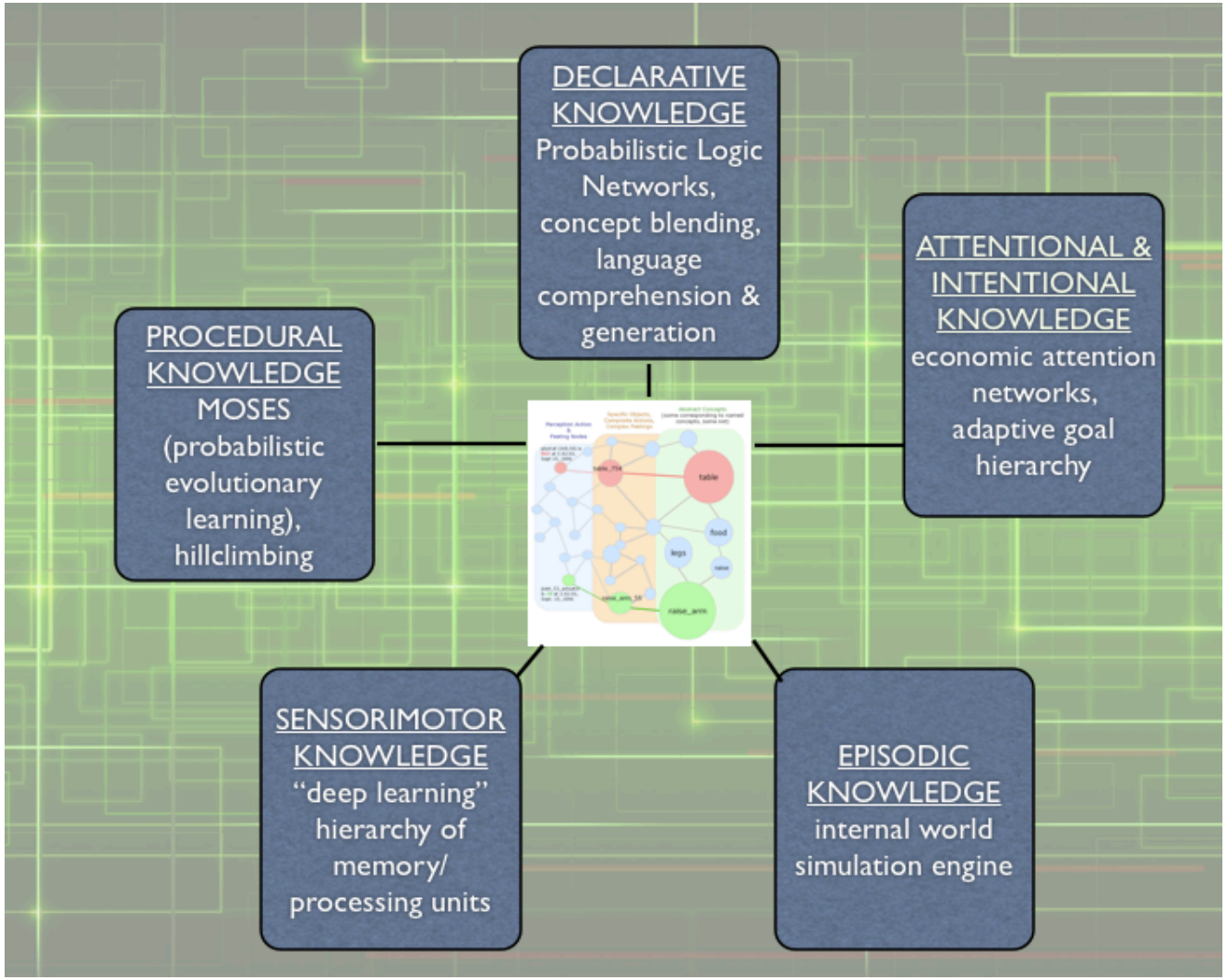

Copyrighted Material
ATLANTIS PRESS
Atlantis Thinking Machines
Series Editor: K.-U. Kühnberger
Ben Goertzel
Cassio Pennachin
Nil Geisweiller
Engineering General Intelligence, Part 2
The CogPrime Architecture for Integrative, Embodied AGI
Copyrighted Material

OpenCog
stores and manipulates knowledge in the form of complex graphs

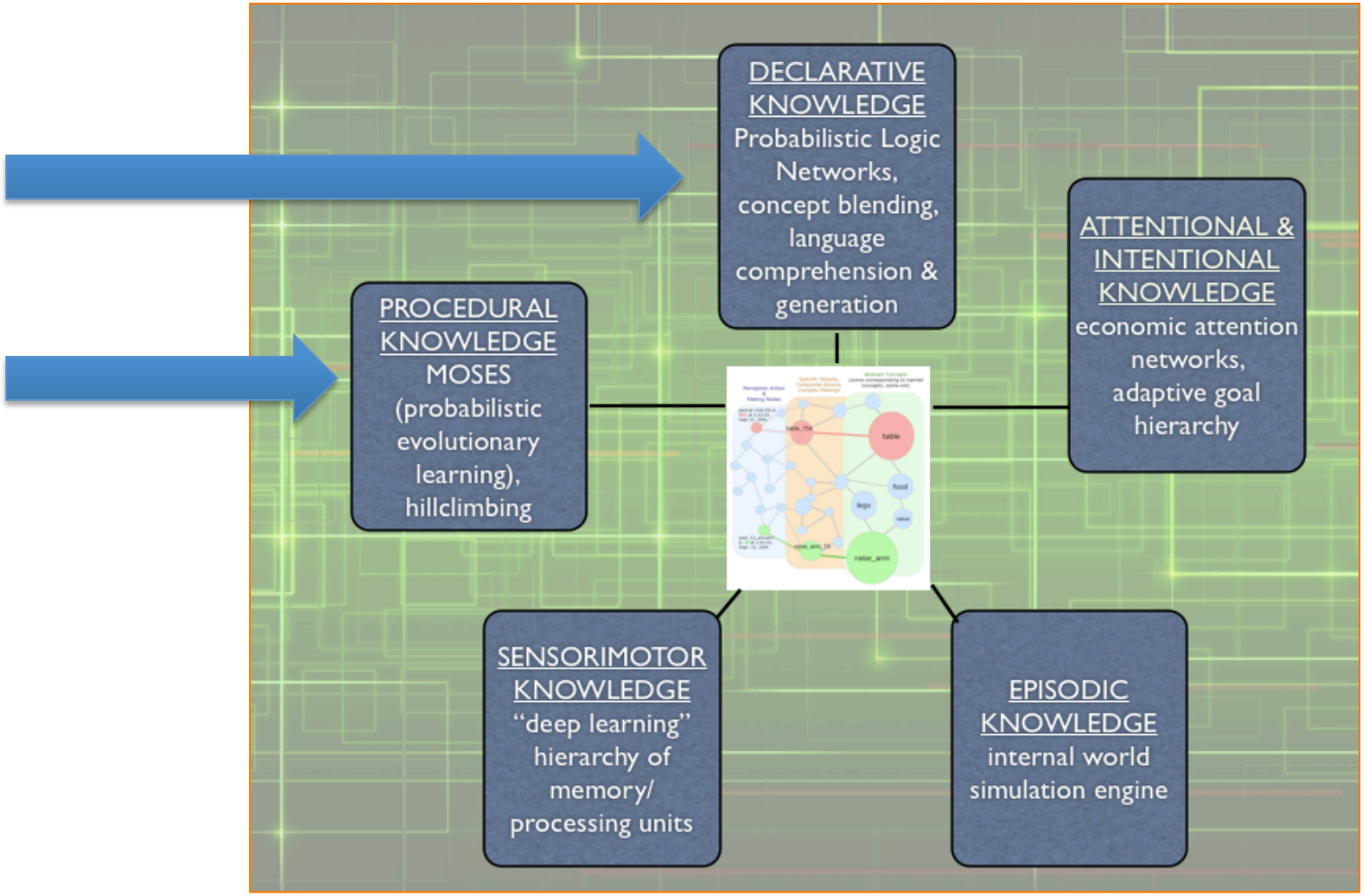
(weighted, labeled hypergraphs)







OpenCog features multiple cognitive algorithms, each acting on different sorts of knowledge within the common "Atomspace" dynamic knowledge store. The aim is to achieve high levels of general intelligence via "cognitive synergy" between the different cognitive algorithms, cooperating together to help an agent choose actions based on goals, context and experience.



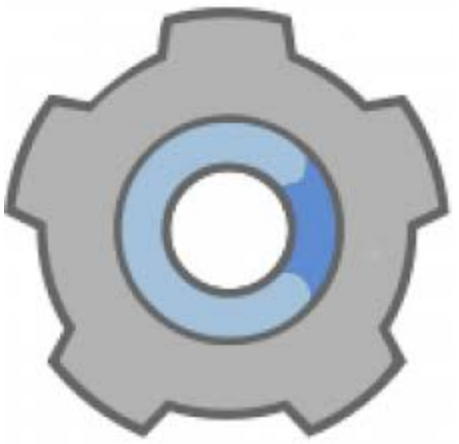
So far, two of OpenCog's cognitive algorithms (MOSES and Probabilistic Logic Networks (PLN) are being used to help understand genomics data. In time, the full integrated OpenCog architecture will be used to serve the role of an "artificial scientist."

OpenCog AI for Genomics: Two Examples

- MOSES for identifying patterns differentiating supercentenarians from healthy ~80 year olds based on SNP combinations
- PLN (probabilistic logic networks) for using bio-ontologies to identify genes indirectly connected to the longevity phenotype, via a combination of genomic data and ontological knowledge

phenotype classification of whole-genome sequenced samples with boolean models derived via MOSES supervised machine learning

(Mike Duncan & Ben Goertzel)



abstract

- A boolean classification function was constructed using a novel supervised machine learning algorithm to categorize healthy from chronically ill geriatric subjects. From an evenly divided sample set of 783 subjects, a population of boolean functions consisting on average of 130 variables was evolved, with a mean out-of-sample accuracy of 0.851, compared to an in-sample accuracy of 0.860.

- The same analysis pipeline was used to distinguish 17 super centenarians from a subset of the above data set consisting of 230 healthy geriatric females. Five significant functions were evolved, four binary and one with a single variable was evolved, with perfect out-of-sample accuracy. These functions consisted of 5 distinct SNP variants.

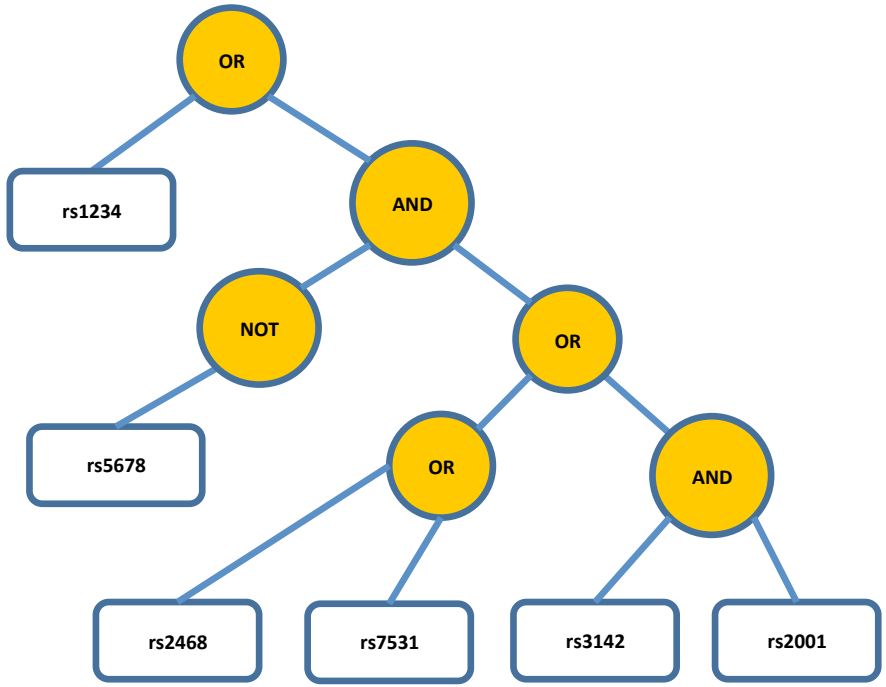
meta optimizing semantic evolutionary search (MOSES)

- MOSES is a 2 level genetic programing algorithm to search catagorization function space, allowing detailed exploration of multiple local fitness maxima.
- Functions from the meta-level population are selected and “mutated” (their neighborhood in function space is searched).
- Variants with improved fitness (better at categorizing) are simplified and returned to the meta-population.
- In addition, integrated feature selection and multiple tunable search and fitness functions improve on standard genetic programing algorithms

epistatic boolean classification models

- MOSES evolves programs coded in a simple programing language called combo.
- Binary variables are valued “0” if a sample is homozygous for the reference allele and “1” for any alternate alleles for a particular variant. A “true” value indicates “case” status.
- an example boolean combo program applied to simulated genomic data:

```
or( $rs1234 and( !$rs5678 or( or( $rs2468 $rs7531 ) and( $rs3142 $rs2001 ))))
```



variable	sample 1	sample 2
rs1234	ref (0)	ref (0)
rs2001	ref (0)	alt (1)
rs2468	ref (0)	ref (0)
rs3142	ref (0)	alt (1)
rs5678	ref (0)	ref (0)
rs7531	ref (0)	lt (1)
program value	control (false)	case (true)

supervised machine learning strategy

- A cross validation strategy is used where the data set is randomly partitioned into training and testing sets at a ratio of 4:1.
- Accuracy scores on training and testing sets are compared for each combo to assess over-fitting.
- Ensembles of combos can be averaged to increase accuracy on out-of-sample data.
- Ranked lists of variants can be constructed by counting variable occurrence in combo ensembles.

whole genome variation data sets

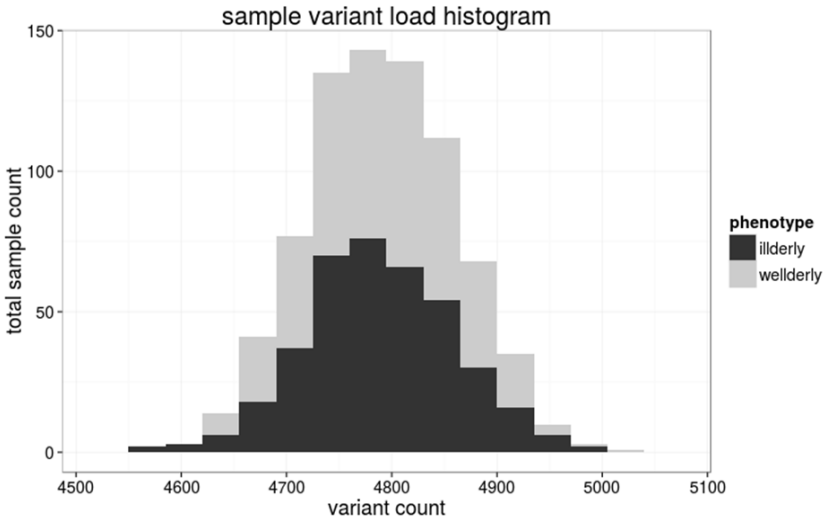
wellderly and illderly data set

- from Scripps
- 783 samples aged 80 and above
- 342 males and 441 females
- 397 wellderly cases and 386 illderly controls
- 230 samples in wellderly female subset

super centenarian data set

- From Stanford
- 17 samples aged 110 and above
- 16 females and 1 male
- 14 whites, 2 Latinas, and 1 African American

wellderly vs. illderly MOSES analysis



- There were 900 combos with accuracies significantly greater than the case prevalence ($p > 0.05$, McNemar's test)
- mean of 130 features per combo
- The mean out-of-sample accuracy of combo ensembles was 0.884.
- Means for all combos in each cross validation set:

	accuracy	precision	recall
mean out-of-sample	0.851	0.863	0.843
mean in-sample	0.860	0.871	0.850
example combo	0.880	0.863	0.909

example combo

variables are gemini db reference ID numbers

and(or(and(or(and(or(and(or(and(or(and(!$X106020 !$X168745) $X763139) !$X735449 !$X297852) and(or($X53710 $X297852 $X552647) !$X766840 $X808350)) or($X14463 $X766840)) and(or(and(or(!$X735449 $X54045) !$X67669) and(or($X135964 !$X808350) $X558377) $X434945) or(and($X735449 $X522743) and($X497883 $X702846) $X431028 !$X480341) or(and($X735449 $X552647) and(!$X135964 !$X194619))) and(or(and($X217849 $X256079 !$X808350) $X297852) !$X735449 $X695782)) or(and(!$X735449 !$X67669 $X434945) !$X427182)) and(or(and(or($X256079 $X808350) $X135964) !$X217849 $X434945) or($X735449 $X766840) $X67669 $X427182) and(or(and($X735449 !$X165425 $X808350) !$X67669 $X434945) or(and($X16581 !$X695782) $X128740 !$X135964 $X434945) !$X14463 $X217849) $X379084) or(and($X14463 $X217849) !$X67669 $X106020 $X379084 !$X427182 !$X480341) or(!$X379084 !$X379090)) and(or(and(or(and($X194619 !$X807580) and($X256079 !$X763139) $X217849 $X427182) or(!$X14463 !$X434945) !$X135964) and($X735449 $X431028) and(!$X106020 $X434945)) or(and(or(!$X165425 ! $X217849) $X558377) and(or($X807580 !$X808350) $X695782) and(!$X735449 !$X14463) and($X480341 !$X807580) $X135964) or(and($X427182 $X497883) !$X67669) or(and(!$X558377 !$X807580) !$X14463 $X53710 $X427182) !$X297852 $X379084)) or(and(or(!$X379090 $X427182) !$X128740) and(!$X67669 $X135964) and(! $X480341 !$X577132) !$X808350))

wellderly vs. illderly top combo SNPs

- The top 25 variants ranked by number of occurrences in the 10000 best combos from 10 cross validation runs.
- "Category" indicates if variable is negated, i.e. if variant is negated in combo then category is "control" because combos are "true" for cases. Note variables can have different categories in different combos.
- Alternate allele frequencies (AAFs) are shown for the data set and 2 reference genome sets: the Exome Aggregation Consortium (ExAC) and the 1000 Genomes.
- Annotations are from gemini[1] v1.7.0 (ensembl v75, dbSNP v141, ExAC v0.3)

1. https://gemini.readthedocs.org/en/latest/index.html

rs id	combo count	category	cyto-band	gene	transcript	data AAF	adjusted ExAC AAF	1k Genomes AAF
rs10953303	2719	control	chr7q22.1	ZAN	ENST00000546213	0.211	0.236	0.198
rs1050348	2206	control	chr6q21	LAMA4	ENST00000389463	0.405	0.665	0.758
rs6942733	2120	case	chr7q22.1	ZAN	ENST00000538115	0.255	0.235	0.199
rs1050348	1308	case	chr6q21	LAMA4	ENST00000389463	0.405	0.665	0.758
rs6942733	1004	control	chr7q22.1	ZAN	ENST00000538115	0.255	0.235	0.199
rs2243191	1002	control	chr1q32.1	IL19	ENST00000270218	0.203	0.748	0.673
rs1688005	901	case	chr19q13.12	FXYD5	ENST00000588699	0.261	0.322	0.412
rs4842978	900	control	chr15q25.2	WDR73	ENST00000561447	0.432	NA	0.726
rs1977420	899	control	chr11p13	APIP	ENST00000395787	0.359	0.404	0.457
rs7905784	802	case	chr10p13	MCM10	ENST00000378694	0.152	0.118	0.064
rs7905784	801	control	chr10p13	MCM10	ENST00000378694	0.152	0.118	0.064
rs1381057	801	control	chr3q13.33	POLQ	ENST00000264233	0.327	0.722	0.745
rs1977420	797	case	chr11p13	APIP	ENST00000395787	0.359	0.404	0.457
rs10953303	720	case	chr7q22.1	ZAN	ENST00000546213	0.211	0.236	0.198
rs2228331	703	case	chr2q37.3	GPC1	ENST00000264039	0.307	0.664	0.664
rs4842978	701	case	chr15q25.2	WDR73	ENST00000561447	0.432	NA	0.726
rs2397084	604	case	chr6p12.2	IL17F	ENST00000336123	0.102	0.069	0.033
rs6587467	603	control	chr1q44	OR2T6	ENST00000355728	0.309	0.721	0.773
rs912174	601	case	chr9p24.3	KANK1	ENST00000382293	0.225	0.219	0.204
rs671694	601	control	chr7p22.2	SDK1	ENST00000404826	0.268	0.752	0.799
rs11895564	600	case	chr2q31.1	ITGA6	ENST00000264106	0.294	0.281	0.252
rs7386783	600	case	chr8q24.22	OC90	ENST00000254627	0.277	0.729	0.737
rs11250	600	control	chr4q13.2	CENPC	ENST00000273853	0.392	0.657	0.701
rs4802648	599	case	chr19q13.33	ZNF473	ENST00000595661	0.221	NA	0.197
rs10277	598	case	chr5q35.3	C5orf45	ENST00000376931	0.433	0.626	0.688

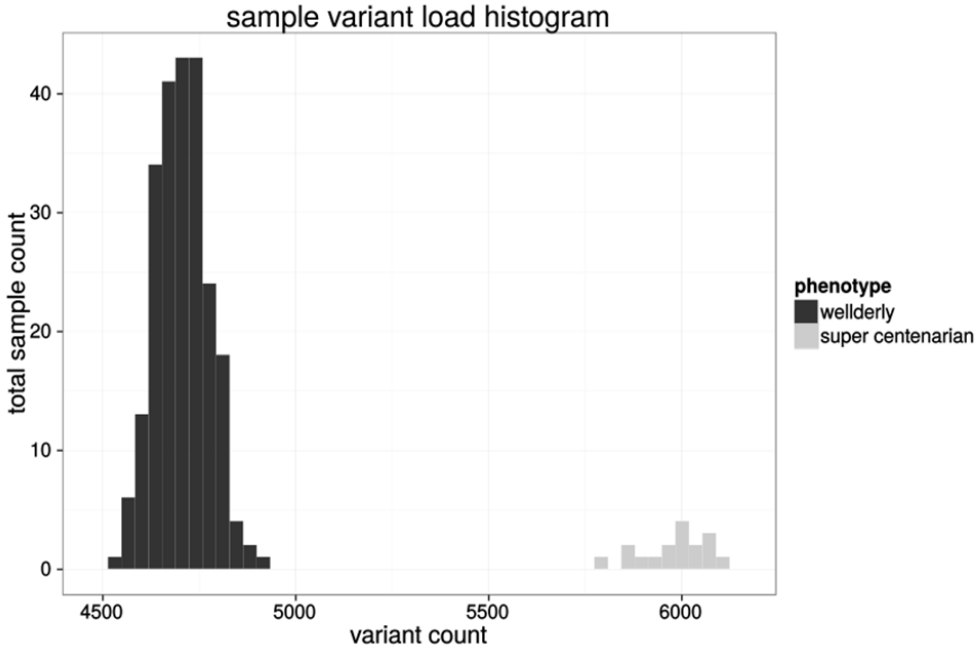
wellderly vs. illderly combo SNPs effects

- Predicted translation effects of variants in top combos
- Selected from feature set of 13,242 SNPs classified in gemini db[1] as “high” and “medium” impact
- Sequence Ontology (SO) impact classification determines gemini impact severity for variant filtering.
- The combined annotation scoring tool (CAROL)[2] combines SIFT[3] and PolyPhen-2[4] nucleotide scores to predict SNP effect on translated protien.

1. https://gemini.readthedocs.org/en/latest/content/database_schema.html#details-of-the-impact-and-impact-severity-columns
2. Lopes MC, Joyce C, Ritchie GRS, John SL, Cunningham F, Asimit J, Zeggini E. A combined functional annotation score for non-synonymous variants Human Heredity (in press)
3. Kumar P, Henikoff S, Ng PC. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm Nature Protocols 4(8):1073-1081 (2009) doi: 10.1038/nprot.2009.86
4. Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR. A method and server for predicting damaging missense mutations Nature Methods 7(4):248-249 (2010)

SNPdb ID	gene	gene name	category	ensembl transcript	SO impact	CAROL prediction
rs1977420	APIP	APAF1 interacting protein	case & control	ENST00000395787	missense variant	Neutral (0.876)
rs10277	C5orf45	chromosome 5 open reading frame 45	case	ENST00000376931	missense variant	Neutral (0.773)
rs11250	CENPC	centromere protein C	control	ENST00000273853	missense variant	Neutral (0.000)
rs1688005	FXYD5	FXYD domain containing ion transport regulator 5	case	ENST00000588699	missense variant	Neutral (0.705)
rs2228331	GPC1	glypican 1	case	ENST00000264039	missense variant	Neutral (0.000)
rs2397084	IL17F	interleukin 17F	case	ENST00000336123	missense variant	Deleterious (1.000)
rs2243191	IL19	interleukin 19	control	ENST00000270218	missense variant	Neutral (0.000)
rs11895564	ITGA6	integrin, alpha 6	case	ENST00000264106	missense variant	Neutral (0.724)
rs912174	KANK1	KN motif and ankyrin repeat domains 1	case	ENST00000382293	missense variant	Neutral (0.000)
rs1050348	LAMA4	laminin, alpha 4	case & control	ENST00000389463	missense variant	Neutral (0.307)
rs7905784	MCM10	minichromosome maintenance complex 10	case & control	ENST00000378694	missense variant	Neutral (0.380)
rs7386783	OC90	otoconin 90	case	ENST00000254627	missense variant	Neutral (0.436)
rs6587467	OR2T6	olfactory receptor, family 2T, member 6	control	ENST00000355728	missense variant	Neutral (0.670)
rs1381057	POLQ	polymerase (DNA directed), theta	control	ENST00000264233	missense variant	Neutral (0.000)
rs671694	SDK1	sidekick cell adhesion molecule 1	control	ENST00000404826	missense variant	Neutral (0.200)
rs4842978	WDR73	WD repeat domain 73	case & control	ENST00000561447	splice region variant	nan
rs10953303	ZAN	zonadhesin (gene/pseudogene)	case & control	ENST00000546213	missense variant	Deleterious (0.999)
rs6942733	ZAN	zonadhesin (gene/pseudogene)	case & control	ENST00000538115	missense variant	Neutral (0.036)
rs4802648	ZNF473	zinc finger protein 473	case	ENST00000595661	splice donor variant	nan

super centenarian vs. wellderly female MOSES input data



- Super centenarian cases were matched to wellderly female controls to attempt to find SNPs associated with extreme longevity.
- Feature set constructed from intersection of SNPs in case & control sets.
- Cases have almost 30% more variants per sample than controls.

super centenarian vs. wellderly female MOSES results

- There were 5 combos with accuracies significantly greater than the case prevalence ($p > 0.05$, McNemar's test)
- The out-of-sample accuracy for all significant combos was 1.0

1. and(!$rs17521570 $rs5905720)
2. and(!$rs2230681 $rs5905720)
3. and(!$rs557337 $rs5905720)
4. and($rs2230681 $rs5905720)
5. $rs5905720

SNPdb ID	gene	gene name	location	transcript	data AAF[1]	ExAC[2] AAF	1kG[3] AAF
rs5905720	MAGIX	MAGI family member, X-linked	chrXp11.23	ENST00000425661	0.021	0.002	0.0003
rs2230739	ADCY9	adenylate cyclase 9	chr16p13.3	ENST00000294016	0.302	0.357	0.260
rs17521570	RAI14	retinoic acid induced 14	chr5p13.2	ENST00000515799	0.116	0.119	0.099
rs2230681	PSMD9	proteasome 26S subunit, non-ATPase 9	chr12q24.31	ENST00000261817	0.123	0.856	0.834
rs557337	TBC1D4	TBC1 domain family, member 4	chr13q22.2	ENST00000377636	0.065	0.083	0.174

[1] Alternate Allele Frequency
[2] Exome Aggregation Consortium
[3] 1000 Genomes

summary

MOSES can find a diverse set of accurate boolean categorization functions even in data with very large feature sets and highly imbalanced sample category sizes.

Probabilistic Logic for Connecting Genomic Data Patterns with Bio-Ontologies: A simple example

(Eddie Monroe & Ben Goertzel)

Logic: very general, flexible framework for carrying out abstract reasoning.

Encompasses both mathematical and commonsense reasoning.

Probability theory: very general, flexible framework for carrying out reasoning based on uncertainty.

Used in a huge variety of areas including data mining, robotics, vision processing, etc.

“**Progic**” = probability + logic

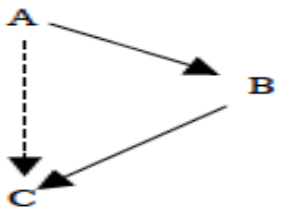
- Various approaches to synthesizing probability and logic exist
- Probabilistic Logic Networks (PLN) is a “progic” framework oriented toward artificial general intelligence.

Probabilistic Logic Networks

- OpenCog represents knowledge in its “Atomspace” in terms of nodes and links of various types
- PLN contains a set of probabilistic logic rules, that transform sets of nodes/links into other sets of nodes/links
- PLN can do deduction, induction, abduction, analogy and other types of reasoning
- PLN can reason on any kind of data, including data-patterns (“combo models”) learned in genomic data by MOSES, or data imported into OpenCog from bio-ontologies
- Due to its ability to process huge amounts of information in subtle ways, PLN can identify data patterns the human mind will miss
- A fundamentally different paradigm than currently popular “machine learning” or “deep learning” architectures, with more capability for abstract symbolic understanding – but can work together with more standard ML algorithms

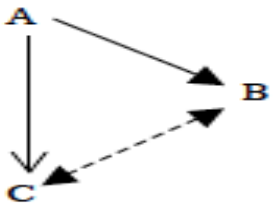
Term Logic

A → B
B → C
|–
A → C



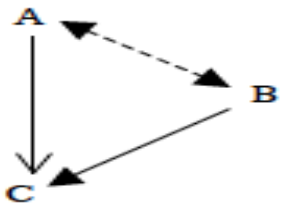
Deduction

A → B
A → C
|–
B → C



Induction

A → C
B → C
|–
A → B



Abduction

Predicate Logic

A

A --> B

|-

B

Multiple PLN Relationship Types

PLN involves more than a dozen logical relationship types, each with particular semantics.

For instance

A → B
B → C
|-
A → C

could be interpreted in many ways including

```
ExtensionalInheritance A B
ExtensionalInheritance B C
|-
ExtensionalInheritance A C
```

```
IntensionalInheritance A B
IntensionalInheritance B C
|-
IntensionalInheritance A C
```

“Higher-Order” PLN

Following Pei Wang’s usage in NARS, in PLN we refer to logic regarding variables or higher-order functions as “higher-order”

```
ImplicationLink
        EvaluationLink has($X, mouth)
        EvaluationLink eats($X, food)
```

Quantifying Truth Values

Each PLN relationship has a truth value attached to it. PLN supports truth value objects of different types, e.g.

- Single probability
- SimpleTruthValue:
  - (s,c) = (probability, confidence level)
  - (s,n) = (probability, amount of evidence)
- Imprecise truth value
  - (L,U) interval, e.g. (.4,.6)
- Indefinite truth value
  - (L,U,b,k) … interval plus confidence level b, and "personality parameter" k, e.g. (.4,.6,.9,2)
- Distributional truth value
  - first or second order pdf

Example PLN rule+formula: deduction

B $<s_B>$
C $<s_C>$
ExtensionalInheritance A B $<s_{AB}>$
ExtensionalInheritance B C $<s_{BC}>$
|-
ExtensionalInheritance A C $<s_{AC}>$
$s_{AC} = s_{AB}\ s_{BC} + (1-s_{AB})\ (\ s_C - s_B\ s_{BC}\ )\ /\ (1-\ s_B\ )$

As given above, this acts on single-probability truth values. It can be extended to other true value forms.

PLN rules

Each rule maps a tuple of relationships into a relationship

Example: deduction rule

```
Subset A B
Subset B C
|-
Subset A C
```

PLN formulas

Each formula maps a tuple of truth values into a truth value

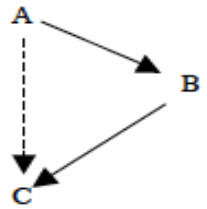
Example: deduction formula

$$s_{AC} = s_{AB}\, s_{BC} + (1-s_{AB})\, (\, s_C - s_B\, s_{BC}\,) \,/\, (1-\, s_B\,)$$

Inversion
(Bayes Rule)

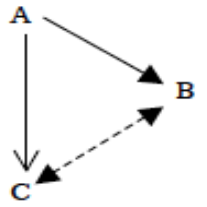
```
A
B
Subset A B
|-
Subset B A
```

A → B
B → C
|-
A → C



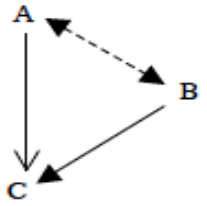
Deduction

A → B
A → C
|-
B → C



Induction

A → C
B → C
|-
A → B



Abduction

In PLN, simple first-order **induction** and **abduction** are obtained by combining deduction and Bayes rule.

More advanced induction and abduction result from using intensional relationships.

Glossary of Link Types

AttractionLink

Indicates the extent to which one concept is a pattern or property helping to characterize another.

(AttractionLink A B) indicates the extent to which B is a property that characterizes A.

ConceptNode

A node representing any concept.

ExecutionOutputLink

Indicates execution of a function with a list arguments to that function. This allows for atomspace representation of the execution of arbitrary code.

GeneNode

A node representing a particular gene.

Glossary of Link Types (cont.)

GroundedSchemaNode

Specifies the name of a predefined procedure that is to be called.

ImplicationLink

Expresses an if...then... relation, or that the truth of one predicate implies the truth of another.

(ImplicationLink A B) denotes that A implies B.

IntensionalEquivalenceLink

Indicates that the properties associated with one predicate being true are similar to the properties associated with another predicate being true.

(IntensionalEquivalanceLink A B) denotes that the properties associated with A being true are similar to the properties associated with B being true.

IntensionalImplicationLink

Expresses an if… then... relation between the *properties* of 2 predicates.

(IntensionalImplicationLink A B) denotes that the properties of A imply the properties of B.

Glossary of Link Types (cont.)

IntensionalSimilarityLink

Indicates that two concepts have similar properties.

(IntensionalSimilarityLink A B) denotes that the properties of A are similar to the properties of B

ListLink

Used for grouping Atoms for some purpose, typically to specify a set of arguments to some function or relation.

MemberLink

Indicates set membership.

(MemberLink x S) denotes that element x is a member of set S. The TruthValue associated with a MemberLink is meant to indicate fuzzy set membership.

NotLink

Corresponds to the negation of a concept or predicate.

Glossary of Link Types (cont.)

PredicateNode

Names the predicate of a relation. Predicates are functions that have arguments and produce a truth value as output.

SetLink

A type of link used to group its arguments into a set

(SetLink x y z) simply indicates that there is a set {x,y,z}

SubsetLink

Denotes extensional inheritance, which is inheritance between sets based on their members. It specifies an “is-an-instance-of” relationship.

(SubsetLink A B) specifies that A is an instance of B.

Example Inference: Goal

- Through MOSES analysis, we found overexpression of LY96 appears to distinguish Nonagenarians from controls.

- Using PLN, what can we infer about the relationship between LY96 and longevity based on background domain and experimental knowledge?

Our target conclusion is:

```
ImplicationLink
  (ExecutionOutputLink
     (GroundedSchemaNode "scm: make-over-expression-predicate")
     (GeneNode ”LY96"))
  (PredicateNode "LongLived”)
```

Interpretation: “Overexpression of LY96 implies longevity.”

Background Information

There is pre-existing evidence that over-expression of gene TBK1 is associated with increased lifespan (Source: Lifespan Observations Database)

```
(IntensionalImplicationLink (stv 0.3 0.7)
  (ExecutionOutputLink (stv 0.2 0.7)
    (GroundedSchemaNode "scm: make-overexpression-predicate")
    (ListLink
      (GeneNode "TBK1" (stv .0004 0.9))))
  (PredicateNode "LongLived" (stv 0.15 0.8)))
)
```

Interpretation: "Overexpression of TBK1 implies longevity"

Genes are associated with Gene Ontology terms and other categories.

```
((MemberLink
 (GeneNode "TBK1" (stv 0004 0.9))
 (ConceptNode "GO:0005515" (stv 0.001 0.9)))

(MemberLink
 (GeneNode "TBK1" (stv 0004 0.9))
 (ConceptNode "GO:0045087" (stv 0.001 0.9)))
. . .
```

Interpretation: "TBK1 is a member of GO category 0005515,"
"TBK1 is a member of GO category 0045087,"
. . . for each gene category annotation

Inference Chain Steps

(1) Member-to-Subset Rule

(Member A B) |- (Subset (Set A) B)

Premises:

```
(MemberLink
  (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998))
  (ConceptNode "GO:0051607" (stv 0.001 0.89999998))
)
. . .
```

“TBK1 is a member of GO category 0051607”

Conclusions:

```
(SubsetLink
  (SetLink
    (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998))
  )
  (ConceptNode "GO:0051607" (stv 0.001 0.89999998))
)
. . .
```

“The singleton set containing TBK1 is a subset of GO category 0051607”

Intensional Similarity

- We will infer a relationship between the gene LY96 and the predicate LongLived through the similarity of LY96 with gene TBK1, which is already known to be related to longevity.
- Intensional similarity is based on common properties of the genes.
- Steps 2-5 that follow are needed for creating the IntensionalSimilarity relationship.

(2) Compare gene properties

- We are using GO category annotations for gene properties.
- At the start of the inference, we need to get the supersets of {TBK1} and {LY96} and determine the intersection and union of the supersets

LY96: member of 25 GO categories

TBK1: member of 34 GO categories

Common categories (intersection):

GO:0005515 protein binding
GO:0045087 innate immune response
GO:0006954 inflammatory response
GO:0010008 endosome membrane
GO:0002224 toll-like receptor signaling pathway
GO:0002756 MyD88-independent toll-like receptor signaling pathway
GO:0007249 I-kappaB kinase/NF-kappaB signaling
GO:0034138 toll-like receptor 3 signaling pathway
GO:0034142 toll-like receptor 4 signaling pathway
GO:0035666 TRIF-dependent toll-like receptor signaling pathway

(3) Subset NotA B Direct Evaluation

(Inheritance A B) |- (Inheritance (Not A) B)

For each common category relationship (LinkType A B), create (LinkType (Not A) B)

Premises:

```
(SubsetLink (stv 1 0.99999982)
  (SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
  (ConceptNode "GO:0045087" (stv 0.001 0.89999998))
)
. . .
```

“{LY96} is a subset of GO:0045087”

Conclusions:

```
(SubsetLink (stv 0.028667862 0.99999982)
  (NotLink
    (SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
  )
  (ConceptNode "GO:0045087" (stv 0.001 0.89999998)))
. . .
```

“A random gene (exclusive of LY96) belongs to GO:0045087 (with a low probability)”

(4) AttractionRule

(And (Subset A B) (Subset (Not A) B)) |- (AttractionLink A B)

Make AttractionLinks for LY96 and TBK1 for each common relationship (IOW for each relationship in the intersection of the supersets).

Premises:

```
(SubsetLink (stv 1 0.99999982)
  (SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
  (ConceptNode "GO:0045087" (stv 0.001 0.89999998))
)
(SubsetLink (stv 0.028667862 0.99999982)
  (NotLink
    (SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
  )
  (ConceptNode "GO:0045087" (stv 0.001 0.89999998)))
. . .
```

{LY96} is a subset of "GO:0045087,"
"A random gene not in {LY96} is a subset of GO:0045087 (with a low probability)"

Conclusions:

```
(AttractionLink (stv 0.97133213 0.99999982)
  (SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
  (ConceptNode "GO:0045087" (stv 0.001 0.89999998)))
. . .
```

"GO:0045087 is a property of/pattern in {LY96)"

(5)IntensionalSimilarity Direct Evaluation

(And (Attraction P A) (Attraction P B) (Attraction (Q A) (Attraction (Q B) …) |- (IntensionalSimilarity A B)

Premises:

(AttractionLink (stv 0.97133213 0.99999982)
(SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
(ConceptNode "GO:0045087" (stv 0.001 0.89999998)))

(AttractionLink (stv 0.97133213 0.99999982)
(SetLink (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998)))
(ConceptNode "GO:0045087" (stv 0.001 0.89999998)))
. . .

"GO:0045087 is a property of {LY96}"
"GO:0045087 is a property of {TBK1}"
Etc. . . .

Conclusion:

(IntensionalSimilarityLink (stv 0.19570713 0.99999982)
(SetLink (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998)))
(SetLink (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))

"{TBK1} properties are similar to {LY96} properties"

(6) Singleton-Similarity-Rule

(Similarity {A} {B}) |- (Similarity A B)

Premise:

```
(IntensionalSimilarityLink (stv 0.19570713 0.99999982)
  (SetLink
    (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998)))
  (SetLink
    (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))
```

"{TBK1} properties are similar to {LY96} properties"

Conclusion:

```
(IntensionalSimilarityLink (stv 0.19570713 0.99999982)
    (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998))
    (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))
  )
```

"TBK1 properties are similar to LY96 properties"

(7) Gene-Similarity-to-Overexpression-Equivalence

(Similarity (Gene A) (Gene B)) |- (Equivalence (A-overexpressed) (B-overexpressed)

Premise:

```
(IntensionalSimilarityLink (stv 0.19570713 0.99999982)
    (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998))
    (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))
  )
```

“TBK1 properties are similar to LY96 properties”

Conclusion:

```
(IntensionalEquivalenceLink (stv 0.19570713 0.99999982)
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998))))
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))))
```

“Properties associated with over-expression of TBK1 are similar to properties associated with overexpression of LY96”

(8) Equivalence-Transformation Rule

(Equivalence A B) |- (And (Implication A B) (Implication B A))

Premise:

```
(IntensionalEquivalenceLink (stv 0.19570713 0.99999982)
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998))))
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "LY96" (stv 4.1666666e-05 0.89999998)))))
```

" 'Overexpression of TBK1 properties' is similar to 'overexpression of RYR1 properties' "

Conclusion:

```
(IntensionalImplicationLink (stv 0.3273496 0.99999982)
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))))
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998)))))
```

"Having properties associated with over-expression of LY96 implies having properties associated with overexpression of TBK1"

(9) Implication Deduction Rule

(And (Implication A B) (Implication B C) |- (Implication A C)
(Part 1)

Premises:

```
(IntensionalImplicationLink (stv 0.3273496 0.99999982)
      (ExecutionOutputLink (stv 0.2 0.69999999)
        (GroundedSchemaNode "scm: make-overexpression-predicate")
        (ListLink
          (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))))
      (ExecutionOutputLink (stv 0.2 0.69999999)
        (GroundedSchemaNode "scm: make-overexpression-predicate")
        (ListLink
          (GeneNode "TBK1" (stv 4.1666666e-05 0.89999998)))))
```

“Having properties associated with overexpression of LY96, implies having properties associated with overexpression of TBK1”

```
(IntensionalImplicationLink (stv 0.3 0.7)
  (ExecutionOutputLink (stv 0.2 0.7)
    (GroundedSchemaNode "scm: make-overexpression-predicate")
    (ListLink
      (GeneNode "TBK1" (stv .0004 0.9))))
  (PredicateNode "LongLived" (stv 0.15 0.8)))
)
```

“Having properties associated with overexpression of TBK1, implies having properties associated with longevity”

(9) Implication Deduction Rule

(And (Implication A B) (Implication B C) |- (Implication A C)
(Part 2)

Conclusion:

```
(IntensionalImplicationLink (stv 0.17387806 0.69999999)
    (ExecutionOutputLink (stv 0.2 0.69999999)
      (GroundedSchemaNode "scm: make-overexpression-predicate")
      (ListLink
        (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))
      )
    )
    (PredicateNode "LongLived" (stv 0.15000001 0.80000001))
  )
```

“Having properties associated with ‘Overexpression of LY96’ implies having properties associated with longevity”

(10) Implication Conversion Rule

(IntensionalImplication A B) |- (Implication A B)

Premise:

```
(IntensionalImplicationLink (stv 0.17387806 0.69999999)
   (ExecutionOutputLink (stv 0.2 0.69999999)
     (GroundedSchemaNode "scm: make-overexpression-predicate")
     (ListLink
       (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))))
   (PredicateNode "LongLived" (stv 0.15000001 0.80000001))
 )
```

"Having properties associated with 'Overexpression of LY96' implies having properties associated with longevity"

Conclusion:

```
(ImplicationLink (stv 0.17387806 0.48999998)
   (ExecutionOutputLink (stv 0.2 0.69999999)
     (GroundedSchemaNode "scm: make-overexpression-predicate")
     (ListLink
       (GeneNode "LY96" (stv 4.1666666e-05 0.89999998))))
   (PredicateNode "LongLived" (stv 0.15000001 0.80000001))
)
```

"Overexpression of LY96 implies longevity" (Our target conclusion)

Next big AI challenge here: Fully automated, scalable inference control (choice of which inference steps to take), via data-mining of inference history

Broad Vision: AI Scientist

- Integrated knowledge-base of all biological (+ chemical etc.) knowledge, in the Atomspace, built in semi-automated way
- Knowledge comes from: datasets, databases, texts, simulations, automated use of lab equipment
- MOSES, PLN and other AI methods used for hypothesis discovery and validation
- Connect OpenCog w/ simulation engine, use OpenCog data/inferences to help set simulation parameters
- AI to design experiments, run robotized experiments
- Language generation to produce written reports
- ***Full-on AI Scientist!!***