



SELVENTA™

Analyze. Accelerate. Clarify.

BEL Framework V1.2

KAM Topology & Compiler Operations

October 2011

Overview

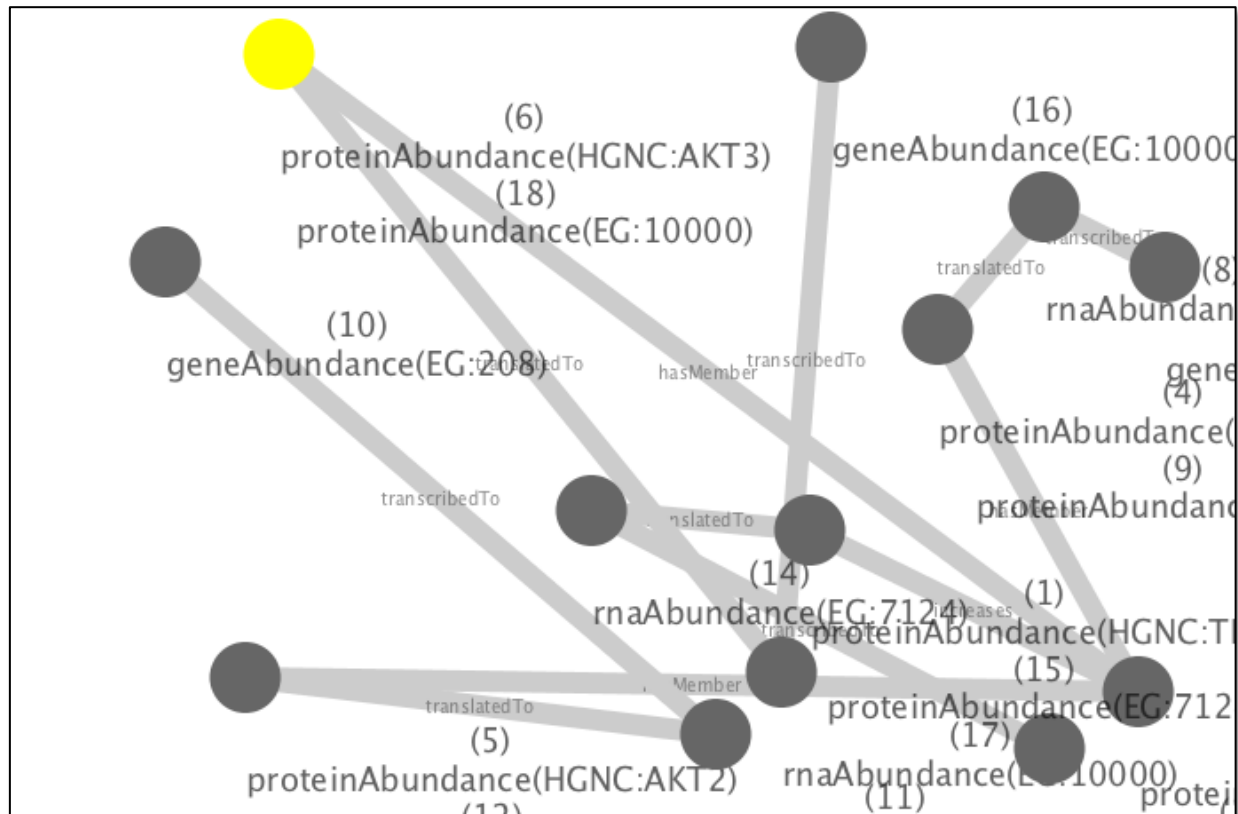
- KAM Structure
- Phase I Compiler Expansions
- Phase III Compiler Augmentations

Kam Structure

- Composed of Nodes (KamNode) and Edges (KamEdge)
- Each KamNode represents one or more BEL Terms drawn from one or more BEL Documents
- Each KamEdge represents one or more BEL Statements from from one or more BEL Documents

KamNodes

- KamNodes are coalesced wherever possible by the equivalencing engine (Phase II)



KamEdges

- Represent assertions supported by one or more BEL Statements
- Querying a KamEdge will return each BEL Statement supporting the assertion
 - Assertions are independent of Annotations, they are coalesced based solely on semantic triple after equivalencing
- Querying a BEL Statement will return
 - The BEL Document the statement was recorded in
 - The list of assertions for the statement

Overview

- KAM Structure
- Phase I Compiler Expansions
- Phase III Compiler Augmentations

Phase I Expansions

- List expansions
- Inner terms
- Protein modifications
- Reactions
- Nested statements

hasMembers Expansion

- Phase I expands **hasMembers** relationships to individual **hasMember** relationships
- All hasMembers relationship statements are removed

`p(PFH:"AKT Family") hasMembers list(p(HGNC:AKT1),p(HGNC:AKT2),p(HGNC:AKT3))`

becomes

`p(PFH:"AKT Family") hasMember p(HGNC:AKT1)`
`p(PFH:"AKT Family") hasMember p(HGNC:AKT2)`
`p(PFH:"AKT Family") hasMember p(HGNC:AKT3)`

hasComponents Expansion

- Phase I expands **hasComponents** relationships to individual **hasComponent** relationships
- All hasComponents relationship statements are removed

```
complex(NCH:"Nfkb Complex") hasComponents \  
  list(p(HGNC:NFKB1),p(HGNC:NFKB2),p(HGNC:REL),p(HGNC:RELA),p(HGNC:RELB))
```

becomes

```
complex(NCH:"Nfkb Complex") hasComponent p(HGNC:NFKB1)  
complex(NCH:"Nfkb Complex") hasComponent p(HGNC:NFKB2)  
complex(NCH:"Nfkb Complex") hasComponent p(HGNC:REL)  
complex(NCH:"Nfkb Complex") hasComponent p(HGNC:RELA)  
complex(NCH:"Nfkb Complex") hasComponent p(HGNC:RELB)
```

complexAbundance Expansion

- Phase I preprocesses **complexAbundance()** terms and injects individual **hasComponent** relationships

`complex(p(HGNC:GTF2E1),p(HGNC:GTF2E2))`

becomes

`complex(p(HGNC:GTF2E1),p(HGNC:GTF2E2))`

`complex(p(HGNC:GTF2E1),p(HGNC:GTF2E2)) hasComponent p(HGNC:GTF2E1)`

`complex(p(HGNC:GTF2E1),p(HGNC:GTF2E2)) hasComponent p(HGNC:GTF2E2)`

compositeAbundance Expansion

- Phase I preprocesses **compositeAbundance()** terms and injects individual **includes** relationships

```
composite(a(CHEBI:"deoxyribonucleic acid"), a(CHEBI:"NAD+")) -> \
  (ribo(p(HGNC:PARP1)) => p(HGNC:PARP1, pmod(R)))
```

becomes

```
composite(a(CHEBI:"deoxyribonucleic acid"), a(CHEBI:"NAD+")) -> \
  (ribo(p(HGNC:PARP1)) => p(HGNC:PARP1, pmod(R)))
composite(a(CHEBI:"deoxyribonucleic acid"), a(CHEBI:"NAD+")) -> \
  (ribo(p(HGNC:PARP1)) includes a(CHEBI:"deoxyribonucleic acid"),
  composite(a(CHEBI:"deoxyribonucleic acid"), a(CHEBI:"NAD+")) -> \
    (ribo(p(HGNC:PARP1)) includes a(CHEBI:"NAD+"))
```

Inner Terms Expansion

- Phase I expands inner terms to relate abundances to activity terms using **actsIn** relationships

$\text{phos}(\text{p(HGNC:DUSP1)}) = | \text{kin}(\text{p(HGNC:MAPK8)})$

becomes

$\text{phos}(\text{p(HGNC:DUSP1)}) = | \text{kin}(\text{p(HGNC:MAPK8)})$

$\text{p(HGNC:DUSP1)} \text{ actsIn } \text{phos}(\text{p(HGNC:DUSP1)})$

$\text{p(HGNC:MAPK8)} \text{ actsIn } \text{kin}(\text{p(HGNC:MAPK8)})$

Protein Modification Expansion

- Phase I expands **proteinModification()** sub-terms to associate a modified protein abundance with the normal (root) protein abundance

$p(\text{HGNC:MAPK1}, \text{pmod}(P, T)) \Rightarrow \text{kin}(p(\text{HGNC:MAPK1}))$

becomes

$p(\text{HGNC:MAPK1}, \text{pmod}(P, T)) \Rightarrow \text{kin}(p(\text{HGNC:MAPK1}))$

$p(\text{HGNC:MAPK1}) \text{ hasModification } p(\text{HGNC:MAPK1}, \text{pmod}(P, T))$

$p(\text{HGNC:MAPK1}) \text{ actsIn } \text{kin}(p(\text{HGNC:MAPK1}))$

Protein Modification Expansion

- Phase I expands **fusion()**, **truncation()**, and **substitution()** sub-terms to associate a protein variant abundance with the normal protein abundance

p(HGNC:KRAS, trunc(55))

becomes

p(HGNC:KRAS, trunc(55))

p(HGNC:KRAS) hasVariant p(HGNC:KRAS, trunc(55))

Reaction Expansion

- Phase I expands **reactants()** and **products()** reaction sub-terms to associate the reactant and product lists with their abundances

```
reaction(reactants(a(CHEBI:superoxide)), \
  products(a(CHEBI:"hydrogen peroxide"),a(CHEBI:oxygen)))
```

becomes

```
reaction(reactants(m(CHEBI:superoxide)), \
  products(a(CHEBI:"hydrogen peroxide"),a(CHEBI:oxygen)))
a(CHEBI:superoxide) reactantIn reaction(reactants(a(CHEBI:superoxide)), \
  products(a(CHEBI:"hydrogen peroxide"),a(CHEBI:oxygen)))
reaction(reactants(a(CHEBI:superoxide)), \
  products(a(CHEBI:"hydrogen peroxide"),a(CHEBI:oxygen))) \
  hasProduct a(CHEBI:"hydrogen peroxide")
reaction(reactants(a(CHEBI:superoxide)), \
  products(a(CHEBI:"hydrogen peroxide"),a(CHEBI:oxygen))) \
  hasProduct a(CHEBI:oxygen)
```

Named Statement Expansion

- The compiler will automatically expand nested statements and create additional relationships from the subject of the statement to the object of the nested statement
 - can be turned off using the `--no-statement-expansion` switch

Default Nested Statement Expansion

- Phase I expands nested statements to link the subject of the statement to the object of the nested statement
- The original statement is preserved as supporting evidence for the derived assertions

```
kin(p(HGNC:MAPK1)) -> \  
  (a(CHEBI:"phorbol 13-acetate 12-myristate") -> p(HGNC:DUSP1))
```

becomes

```
a(CHEBI:"phorbol 13-acetate 12-myristate") -> p(HGNC:DUSP1)  
kin(p(HGNC:MAPK1)) -> p(HGNC:DUSP1)  
p(HGNC:MAPK1) actsIn kin(p(HGNC:MAPK1))
```

Modified Nested Statement Expansion

- When the `–no-statement-expansion` switch is set, the compiler will instantiate the subject of the statement and expand the nested statement but not couple the two together.
- The original statement is removed

```
kin(p(HGNC:MAPK1)) -> \  
  (a(CHEBI:"phorbol 13-acetate 12-myristate") -> p(HGNC:DUSP1))
```

becomes

```
a(CHEBI:"phorbol 13-acetate 12-myristate") -> p(HGNC:DUSP1)  
kin(p(HGNC:MAPK1))  
p(HGNC:MAPK1) actsIn kin(p(HGNC:MAPK1))
```

Overview

- KAM Structure
- Phase I Compiler Expansions
- Phase III Augmentations

Phase III Augmentations

- Gene Activation Pathways
- Protein Family
- Named Complexes

Gene Activation Pathways

- Default behavior it to insert **p()**, **r()**, and **g()** nodes and corresponding edges wherever a protein, rna, or gene abundance term is detected
- The compiler will only insert missing nodes and edges
 - Can be turned off with the **--no-gene-scaffolding** switch

`p(HGNC:KRAS, sub(G, 12, V)) -> path(MESH:Neoplasms)`

becomes

`p(HGNC:KRAS, sub(G, 12, V)) -> path(MESH:Neoplasms)`

`p(HGNC:KRAS) hasVariant p(HGNC:KRAS, sub(G, 12, V))`

`r(HGNC:KRAS) >> p(HGNC:KRAS)`

`g(HGNC:KRAS) := r(HGNC:KRAS)`

Protein Family Expansion

- The compiler will automatically include protein family members when a protein family term is identified
 - Can be turned off using the `--no-protein-families` switch
- The compiler can also search for protein families to include when a protein family member is identified
 - Can be enabled using the `--expand-protein-families` switch
- The compiler will automatically connect protein family activity terms with the corresponding family member activity terms

Protein Family Example 1

$p(\text{HGNC:KRAS}, \text{sub}(\text{G}, 12, \text{D})) \rightarrow \text{kin}(\text{PFH: "MAPK JNK Family"})$

becomes

$p(\text{HGNC:KRAS}, \text{sub}(\text{G}, 12, \text{D})) \rightarrow \text{kin}(\text{PFH: "MAPK JNK Family"})$

$p(\text{HGNC:KRAS}) \text{ hasVariant } p(\text{HGNC:KRAS}, \text{sub}(\text{G}, 12, \text{D}))$

$p(\text{PFH: "MAPK JNK Family"}) \text{ actsIn } \text{kin}(\text{PFH: "MAPK JNK Family"})$

$p(\text{PFH: "MAPK JNK Family"}) \text{ hasMember } p(\text{HGNC:MAPK8})$

$p(\text{PFH: "MAPK JNK Family"}) \text{ hasMember } p(\text{HGNC:MAPK9})$

$p(\text{PFH: "MAPK JNK Family"}) \text{ hasMember } p(\text{HGNC:MAPK10})$

using the default behavior.

Gene activation pathways would then be applied to $p(\text{HGNC:KRAS})$, $p(\text{HGNC:MAPK8})$, $p(\text{HGNC:MAPK9})$, and $p(\text{HGNC:MAPK10})$

Protein Family Example 2

`kin(p(HGNC:AKT1)) -> p(HGNC:RELA)`

becomes

`kin(p(HGNC:AKT1)) -> p(HGNC:RELA)`

`p(HGNC:AKT1) actsIn kin(p(HGNC:AKT1))`

`p(PFH:"AKT Family") hasMember p(HGNC:AKT1)`

`p(PFH:"AKT Family") hasMember p(HGNC:AKT2)`

`p(PFH:"AKT Family") hasMember p(HGNC:AKT3)`

using the `--expand-protein-families` compiler switch.

Gene activation pathways would then be applied to `p(HGNC:AKT1)`, `p(HGNC:AKT2)`, `p(HGNC:AKT3)`, and `p(HGNC:RELA)`

Protein Family Example 3

kin(p(HGNC:AKT1)) -> p(HGNC:RELA)
 kin(p(PFH:"AKT Family")) =| bp(GO:apoptosis)

becomes

kin(p(HGNC:AKT1)) -> p(HGNC:RELA)
 kin(p(PFH:"AKT Family")) =| bp(GO:apoptosis)
 p(HGNC:AKT1) actsIn kin(p(HGNC:AKT1))
 p(PFH:"AKT Family") actsin kin(p(PFH:"AKT Family"))
 p(PFH:"AKT Family") hasMember p(HGNC:AKT1)
 p(PFH:"AKT Family") hasMember p(HGNC:AKT2)
 p(PFH:"AKT Family") hasMember p(HGNC:AKT3)
 kin(p(HGNC:AKT1)) isA kin(p(PFH:"AKT Family"))

Gene activation pathways would then be applied to p(HGNC:AKT1), p(HGNC:AKT2), p(HGNC:AKT3), and p(HGNC:RELA)

Named Complex Expansion

- The compiler will automatically include named complex components when a named complex member is identified
 - can be turned off using the `--no-named-complexes` switch
- The compiler can also search for named complexes to include when a named complex member is identified
 - Can be enabled using the `--expand-named-complexes` switch

Named Complex Example 1

`p(HGNC:NFKBIA) -| tscript(complex(NCH:"Nfkb Complex"))`

becomes

`p(HGNC:NFKBIA) -| tscript(complex(NCH:"Nfkb Complex"))`
`complex(NCH:"Nfkb Complex") actsIn tscript(complex(NCH:"Nfkb Complex"))`
`complex(NCH:"Nfkb Complex") hasComponent p(HGNC:NFKB1)`
`complex(NCH:"Nfkb Complex") hasComponent p(HGNC:NFKB2)`
`complex(NCH:"Nfkb Complex") hasComponent p(HGNC:REL)`
`complex(NCH:"Nfkb Complex") hasComponent p(HGNC:RELA)`
`complex(NCH:"Nfkb Complex") hasComponent p(HGNC:RELB)`

Gene activation pathways would then be applied to `p(HGNC:NFKBIA)`, `p(HGNC:NFKB1)`, `p(HGNC:NFKB2)`, `p(HGNC:REL)`, `p(HGNC:RELA)`, and `p(HGNC:RELB)`

Named Complex Example 2

$\text{kin}(\text{p(HGNC:CHUK)}) \Rightarrow \text{p(HGNC:NFKBIA, pmod(P, S, 32))}$

becomes

$\text{kin}(\text{p(HGNC:CHUK)}) \Rightarrow \text{p(HGNC:NFKBIA, pmod(P, S, 32))}$

$\text{p(HGNC:CHUK) actsIn kin(p(HGNC:CHUK))}$

$\text{p(HGNC:NFKBIA) hasModification p(HGNC:NFKBIA, pmod(P, S, 32))}$

$\text{complex(NCH:"IkappaB Kinase Complex") hasComponents p(HGNC:CHUK)}$

$\text{complex(NCH:"IkappaB Kinase Complex") hasComponents p(HGNC:IKBKB)}$

$\text{complex(NCH:"IkappaB Kinase Complex") hasComponents p(HGNC:IKBKG)}$

Gene activation pathways would then be applied to p(HGNC:CHUK) , p(HGNC:NFKBIA) , p(HGNC:IKBKB) , and p(HGNC:IKBKG)

Network Augmentation Order

