

Modifications

The following functions represent covalent modifications or sequence variation of protein, RNA, or gene abundances. These modifications are special functions that can only be used as an argument within an abundance function.

- Covalent modifications
 - `proteinModification()`, `pmod()`
- Sequence Variations
 - `substitution()`, `sub()`
 - `truncation()`, `trunc()`
 - `fusion()`, `fus()`

Covalent modifications

`proteinModification()`, `pmod()`

The `proteinModification()` or `pmod()` function can be used only as an argument within a `proteinAbundance()` function to indicate covalent modification of the specified protein. Covalently modified protein abundance term expressions have the form:

```
p(ns:v, pmod(<type>, <code>, <pos>))
```

Where `<type>` is one of a set of 9 covalent protein modification types, `<code>` is one of the 20 single-letter amino acid codes, and `<pos>` is the position at which the modification occurs based on the reference sequence for the protein. If `<pos>` is omitted, then the position of the modification is unspecified. If both `<code>` and `<pos>` are omitted, then the residue and position of the modification are unspecified.

An example of a protein modification code would be "P", denoting phosphorylation:

```
p(HGNC:AKT1, pmod(P, S, 21))
```

defines the abundance of human AKT1 phosphorylated at serine 21.

```
p(HGNC:AKT1, pmod(P, S))
```

defines the abundance of human AKT1 phosphorylated at an unspecified serine.

```
p(HGNC:AKT1, pmod(P))
```

defines the abundance of human AKT1 with unspecified phosphorylation.

The following modification types are supported:

Type	Modification
P	Phosphorylation
A	Acetylation
F	Farnesylation
G	Glycosylation
H	Hydroxylation
M	Methylation
R	Ribosylation
S	Sumoylation

U	Ubiquitination
---	----------------

The following single-letter Amino Acid codes are supported:

Code	Amino Acid
A	Alanine
R	Arginine
N	Asparagine
D	Aspartic Acid
C	Cysteine
E	Glutamic Acid
Q	Glutamine
G	Glycine
H	Histidine
I	Isoleucine
L	Leucine
K	Lysine
M	Methionine
F	Phenylalanine
P	Proline
S	Serine
T	Threonine
W	Tryptophan
Y	Tyrosine
V	Valine

Sequence Variations

substitution(), sub()

The `substitution()` or `sub()` function can be used only as an argument within a `proteinAbundance()` function to indicate amino acid substitution of the specified protein, generally resulting from a missense polymorphism or mutation in the corresponding gene. Expressions indicating the abundance of proteins with amino acid substitution sequence variants have the form:

```
p(ns:v, sub(<code_reference>, <pos>, <code_variant>))
```

Where `<pos>` is the position at which the substitution occurs based on the reference sequence for the protein, `<code_reference>` is one of the single-letter amino acid codes and specifies the amino acid at that position in the reference sequence for the protein and `<code_variant>` specifies the amino acid at that position in the variant sequence for the protein.

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

defines the abundance of human KRAS in which a glycine is substituted with valine at codon 12.

truncation(), trunc()

The `truncation()` or `trunc()` function can only be used as an argument within a `proteinAbundance()` function to indicate a truncated protein, generally resulting from a gene sequence variation like a frame shift or nonsense mutation. Expressions indicating the abundance of proteins with truncation sequence variants have the form:

```
p(ns:v, truncation(<pos>))
```

Where `<pos>` is the position at which the truncation occurs based on the reference sequence for the protein,

```
p(HGNC:KRAS, truncation(55))
```

defines the abundance of human KRAS truncated at position 55.

fusion(), fus()

Expressions indicating the abundance of genes, proteins, and rna with fusion modifications have the form: `x(ns1:v1, fus(ns2:v2, a, b))`

Where `x` is either a `proteinAbundance`, a `geneAbundance` or a `rnaAbundance`, `ns1:v1` is the 5' (left side) partner gene and `ns2:v2` is the 3' (right side) partner gene, `a` and `b` are the breakpoints for the 5' and 3' genes respectively.

If `a` and `b` are omitted, the position of the 5' and 3' breakpoints are unspecified.

The following example of a fusion modification:

```
g(HGNC:TMPRSS2, fusion(HGNC:ERG, 365, 38))
```

defines the abundance of human TMPRSS2-ERG fusion gene which encodes nucleotide 1-365 of TMPRSS2 fused to nucleotide 38-3097 of ERG.

If the breakpoint were unknown or unspecified the fusion gene would be represented as:

```
g(HGNC:TMPRSS2, fusion(HGNC:ERG))
```