2011-05-20

1 BEAST

I have added these classes:

`AlloppSpeciesBindingsApSpInfoParser`

`AlloppSpeciesBindingsIndividualParser`

`AlloppSpeciesBindingsParser`

`AlloppSpeciesNetworkModelParser`

`AlloppSpeciesBindings`

`AlloppSpeciesNetworkModel`

`AlloppSpeciesTreeModel - now AlloppLeggedTree`

At the moment, can parse some bits of XML, reading in info about species, individuals and sequences (=taxa) to make a `AlloppSpeciesBindings`.

Then read in XML for `AlloppSpeciesNetworkModel` which jusst contains a parameter for population sizes. This only works for the one tetratree, no diploid tree case, and where there are 2 or 3 species. The node times are just 1.0 and 2.0, and the hybridization times unset.

I have code for translating a network representation into a mullab representation, which is tested by a small case in a test unit `AlloppSpeciesNetworkModelTest`. That does 3 tetra species, and 2 diploid species, in various arrangements. This isn't very relevant to the tetra only case, but it will be needed. I have not tested what it does in the tetra only case.

1.1 XML Parsers

`AlloppSpeciesBindingsApSpInfoParser`, `AlloppSpeciesBindingsIndividualParser`, `AlloppSpeciesBindingsParser`, `AlloppSpeciesNetworkModelParser`

I have copied and adapted code from other parsers. I don't have a good understanding of how they work. I have added `getSyntaxRules()` implementations, but it is hard to tell if I have done as much as I should or could.

1.2 `AlloppSpeciesBindings`

Subclasses

`Individual` - for one individual, one or more sequences

`ApSpInfo` - One (allopolyploid) species, containing one or more individuals

`GeneTreeInfo` - A gene tree as used by BEAST, plus popfactor, plus indices for each individual to map sequences to indices (0 or 1 for tetraploids) which identify the legs of the tetraploid subtree.

[Aside: in cases where both parental diploid species are extinct, there is only one leg, so what exactly is identified here? It matters whether two genes are assigned the same index, or a different one but not what the indices are.]

`GeneTreeInfo.SpeciesSequenceIndices` - where the indices just mentioned are stored.

`GeneTreeInfo.GeneUnionNode` - for `GeneUnionTree`.

`GeneTreeInfo.GeneUnionTree` - serves same function as JH's `CoalInfo`, storing the set (=union) of species-sequence pairs belonging to a node in the gene tree. I copy gene tree topology and times into a `GeneUnionTree` to do calcualtions.

Methods

A lot of mapping of one kind of index or indices to others, one to get list of species at a given ploidy level, etc.

Also `fitsInNetwork()` which visits nodes in gene trees (as `GeneUnionTree`'s) and calls `CoalescenceIsCompatible()` from `AlloppSpeciesNetworkModel`.

1.3 `AlloppSpeciesNetworkModel`

This contains an `AlloppSpeciesBindings`, and two representations of the network:

1. As a 2D array of `AlloppSpeciesTreeModel (=AlloppLeggedTree)`'s. First index is ploidylevel, second is index of (sub)tree within ploidylevel

2. As a multiply labelled binary tree.

Subclasses

`MulLabNode` - for MulLabTree

`MulLabTree` - the multiply labelled binary tree.

`MulLabTree.LegLink` - for conversion of network made of various trees joined together into a multiply labelled binary tree.

`MulLabTree.FootLinks` - for conversion of network made of various trees joined together into a multiply labelled binary tree.

Methods

`makeIntialOneTetraTreeNetwork()` - called by a constructor.

`makesmallexample()` - called by test code.

`CoalescenceIsCompatible()` - not written

`The MulLabTree()` constructor is complicated and calls `simpletree2mullabtree()`, `nodeofunion()`

`MulLabTreeAsNewick()` - for testing.

`numberofpopparameters()` - calculates the number of points in the network where populations are estimated.

1.4 AlloppSpeciesTreeModel (=AlloppLeggedTree)

This is almost a subclass of `AlloppSpeciesNetworkModel` - or `MulLabTree` almost deserves to be top level. It implements a tree with legs.

Subclass

`Leg` - joins root of tree to branch in a lower ploidy tree.

Methods

I have two constructors: for tetraploid-only case; and for small example for testing network to mullab tree conversion.

There are loads of methods delegating to `SimpleTree`.

2 XML

I have edited the TrialAlloppApr18.xml file so it contains bits for the parsers to read