

```

;; The grammar.
;; For Kimura's two-parameter model, the concept of a phylo-grammar
;; is a bit superfluous, but necessary "boilerplate code" to do this
;; sort of thing in xrate.
(grammar
  (name KimuraTwoParameterModel)

  ;; Transformation rules. These follow the pattern for a null model
  ;; with rate matrix X.
  ;; There is one emit state, corresponding to emissions from matrix X.
  (transform (from (S)) (to (X S*)))
  ;; A hacky (but common) way of conditioning on the observed alignment
  ;; length is to set both transition probs from the emit state to one:
  (transform (from (S*)) (to (S)) (prob 1))
  (transform (from (S*)) (to ()) (prob 1))
  ;; Finally we clear a flag, indicating we don't want to re-estimate
  ;; the rule probabilities during EM training:
  (update-rules 0)

  ;; Here are the parameters for Kimura's model.
  (params
    ((alpha 4) ;; transition rate
     ((beta 1) ;; transversion rate
      ) ;; end params

  ;; Now here is the algebraic structure of the rate matrix.
  (chain
    ;; The state of this chain is a single symbol from alphabet DNA.
    ;; Call this symbol X.
    (terminal (X))
    ;; The following line indicates that the initial probabilities
    ;; and mutation rates should be treated as fixed parametric functions,
    ;; not free variables.
    (update-policy parametric)

    ;; initial probability distribution
    (initial (state (a)) (prob 0.25))
    (initial (state (c)) (prob 0.25))
    (initial (state (g)) (prob 0.25))
    (initial (state (t)) (prob 0.25))

    ;; mutation rates
    (mutate (from (a)) (to (c)) (rate beta))
    (mutate (from (a)) (to (g)) (rate alpha))
    (mutate (from (a)) (to (t)) (rate beta))
    (mutate (from (c)) (to (a)) (rate beta))
    (mutate (from (c)) (to (g)) (rate beta))
    (mutate (from (c)) (to (t)) (rate alpha))
    (mutate (from (g)) (to (a)) (rate alpha))
    (mutate (from (g)) (to (c)) (rate beta))
    (mutate (from (g)) (to (t)) (rate beta))
    (mutate (from (t)) (to (a)) (rate beta))
    (mutate (from (t)) (to (c)) (rate alpha))
    (mutate (from (t)) (to (g)) (rate beta))
  ) ;; end chain X
) ;; end grammar

```

```

;; Define the standard DNA alphabet with IUPAC degeneracies
(alphabet
  (name DNA)
  (token (a c g t))
  (complement (t g c a))
  (extend (to n) (from a) (from c) (from g) (from t))
  (extend (to x) (from a) (from c) (from g) (from t))
  (extend (to u) (from t))
  (extend (to r) (from a) (from g))
  (extend (to y) (from c) (from t))
  (extend (to m) (from a) (from c))
  (extend (to k) (from g) (from t))
  (extend (to s) (from c) (from g))
  (extend (to w) (from a) (from t))
  (extend (to h) (from a) (from c) (from t))
  (extend (to b) (from c) (from g) (from t))
  (extend (to v) (from a) (from c) (from g))
  (extend (to d) (from a) (from g) (from t))
  (wildcard *)
) ;; end alphabet DNA

```