The grammar is it currently stands has built been steadily built as more features have been added. To test that the grammar is working properly at each stage, I have created several codes that translate up and down the grammar levels. As it currently stands, the code that generate random stk file only generates basic alphabers, so these translating codes are necessary to generate random stk files with more advanced alphabets,

The original grammar (L0) contains only e (exon),i (intron) and x (intergenic). To this were added transitions: s (start), t (stop) a (acceptor), d (donor), f (remain exon), j (remain intron), y (remain intergenic.) Then a duplicate (g,k,u,v,b,c,h,l) was added to represent the reverse strand. Before moving onto the next stage (L1) some of the letters were changed. This is the newL0 alphabet.

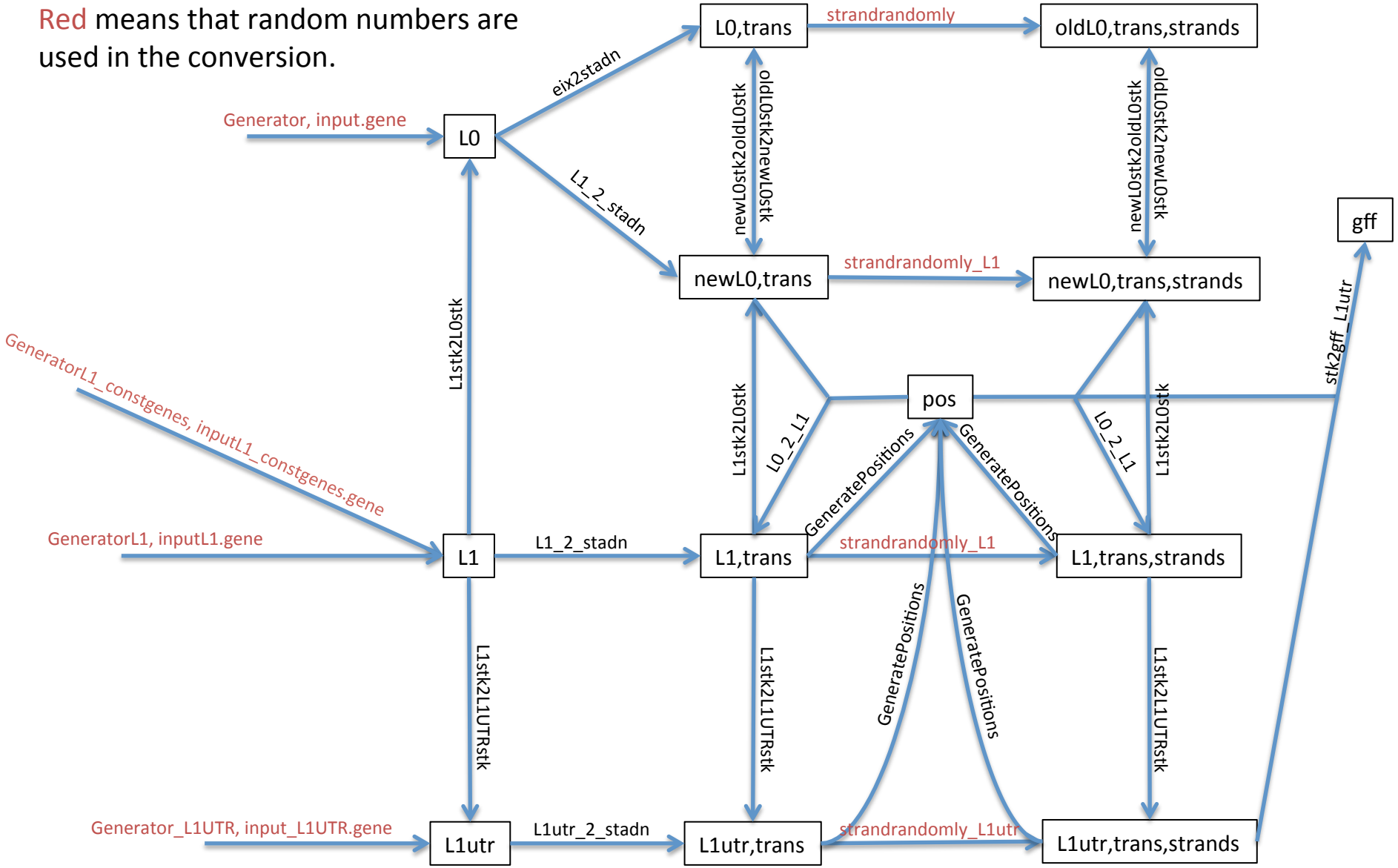
The next stage (L1) adds exon reading frame (e,f,g.) Additional intron (i,j,k), donor (d,o,n) and acceptor (a,b,c) states are added to make sure that reading frame is preserved after splicing. 3 additional remain exon letters (p,q,r) and 3 addition remain intron letters (u,v,w) are also necessary to retain reading frame. Capital letters are used for the reverse strand.

The final stage (L1utr) adds UTR exons (5,3) introns (h,m), donors (~,z), acceptors (|,?), transitional start site ({), transition termination sites (}) and remain UTR transition (6,l,4,y). (Remain intergenic is switched from y to X.) For those characters that cannot be capitalized it uses use the ascii character 15 positions below for the reverse strand if the ascii decimamal is less than 65(A) and 32 positions lower otherwise.

Not yet implemented is the L2 alphabet which will prevent stop sites from being spliced in. This will require three new donor sites for each strand: D1t, D2ta, D2tg (where the letters at the end are the partial reading frame at the end of the exon preceding the donor site.) To keep track of which donor we need corresponding intron states. Finally we need three new acceptor site: A1aa,ag,ga, A2a, A2g. This makes for a total of 88 characters or every printable character that xrate will accept, leaving none for evidence sources that don’t specify an exact letter.

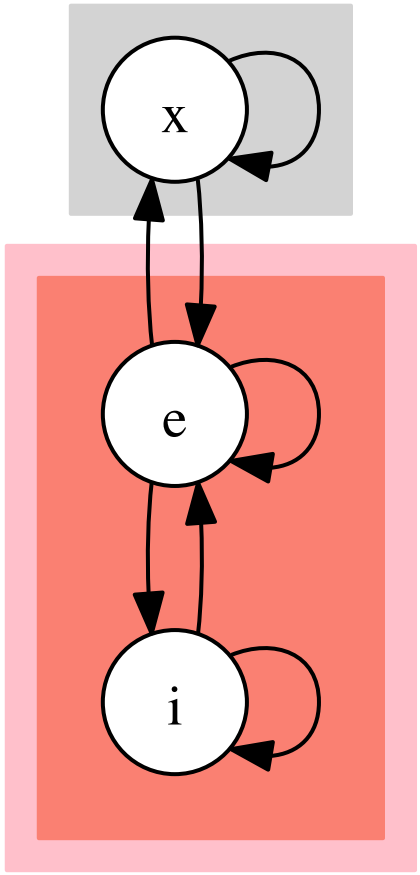
See the rest of the document for a full despcription.

This chart shows which codes convert between which alphabets.
Red means that random numbers are used in the conversion.

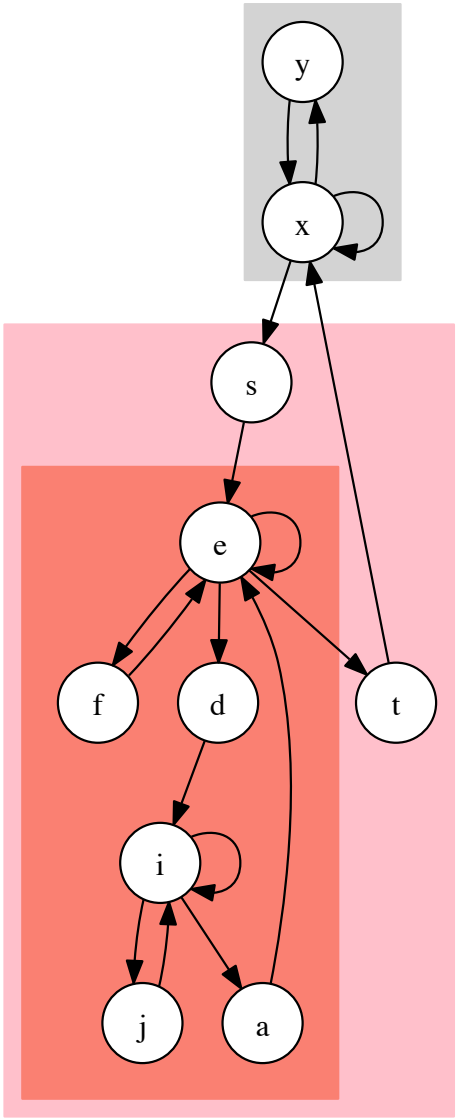


The following charts show which transitions are allowed in each grmmar.

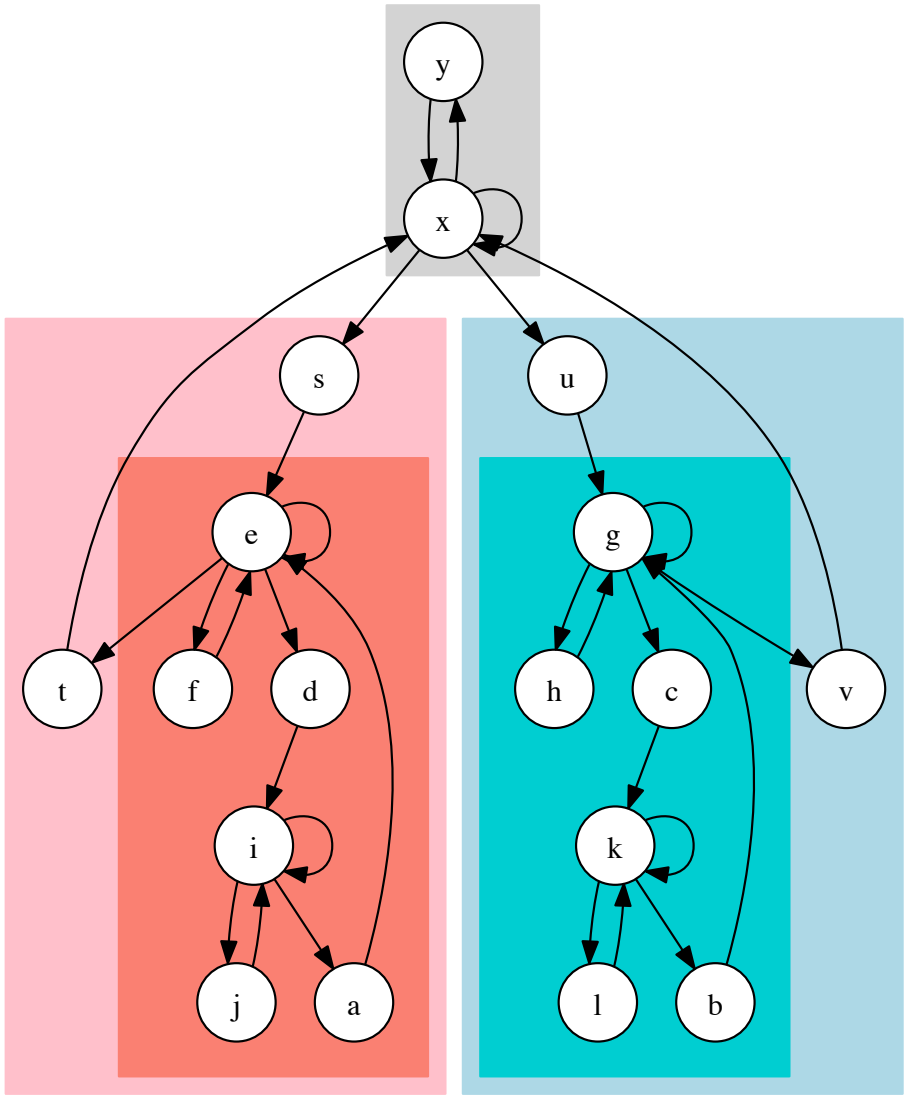
L0



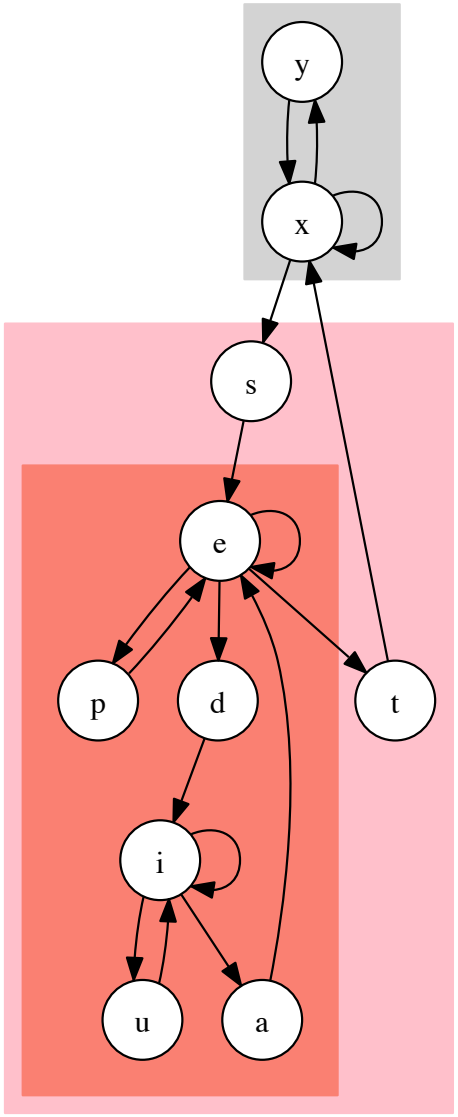
L0,trans



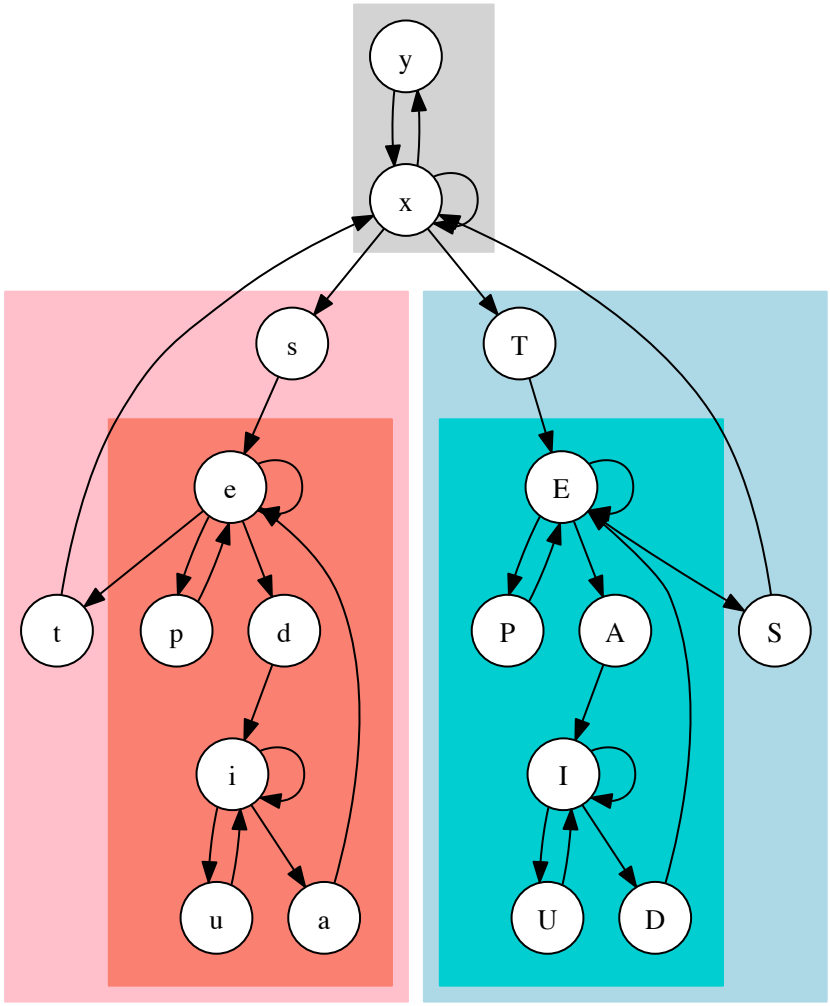
L0,trans,strands



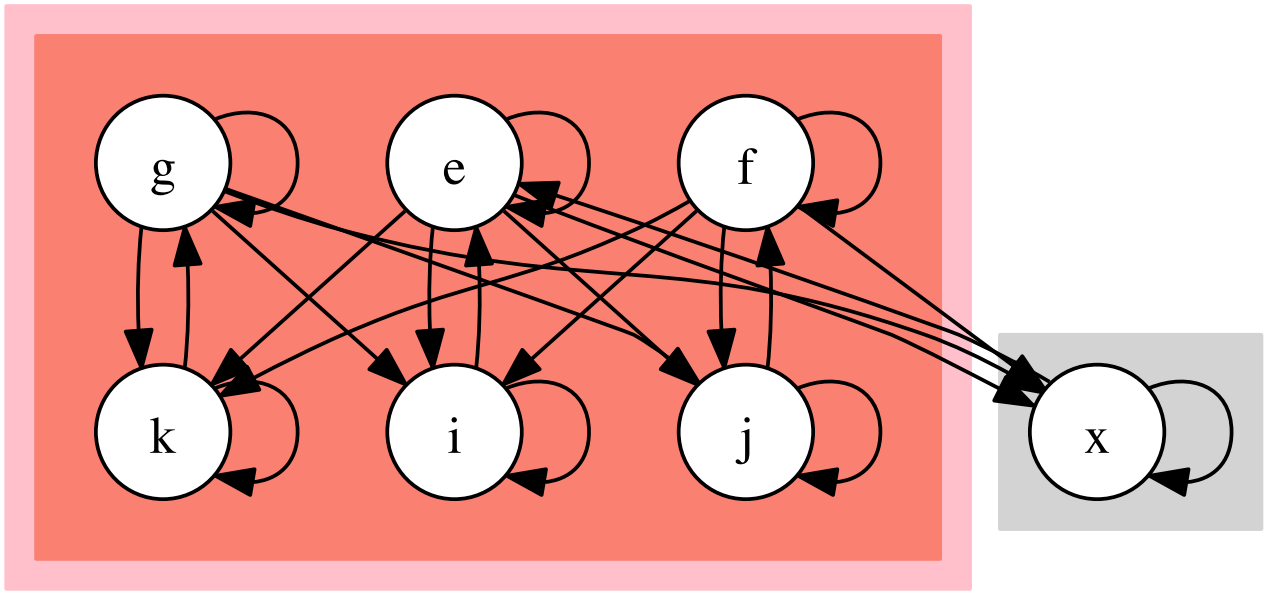
newL0,trans



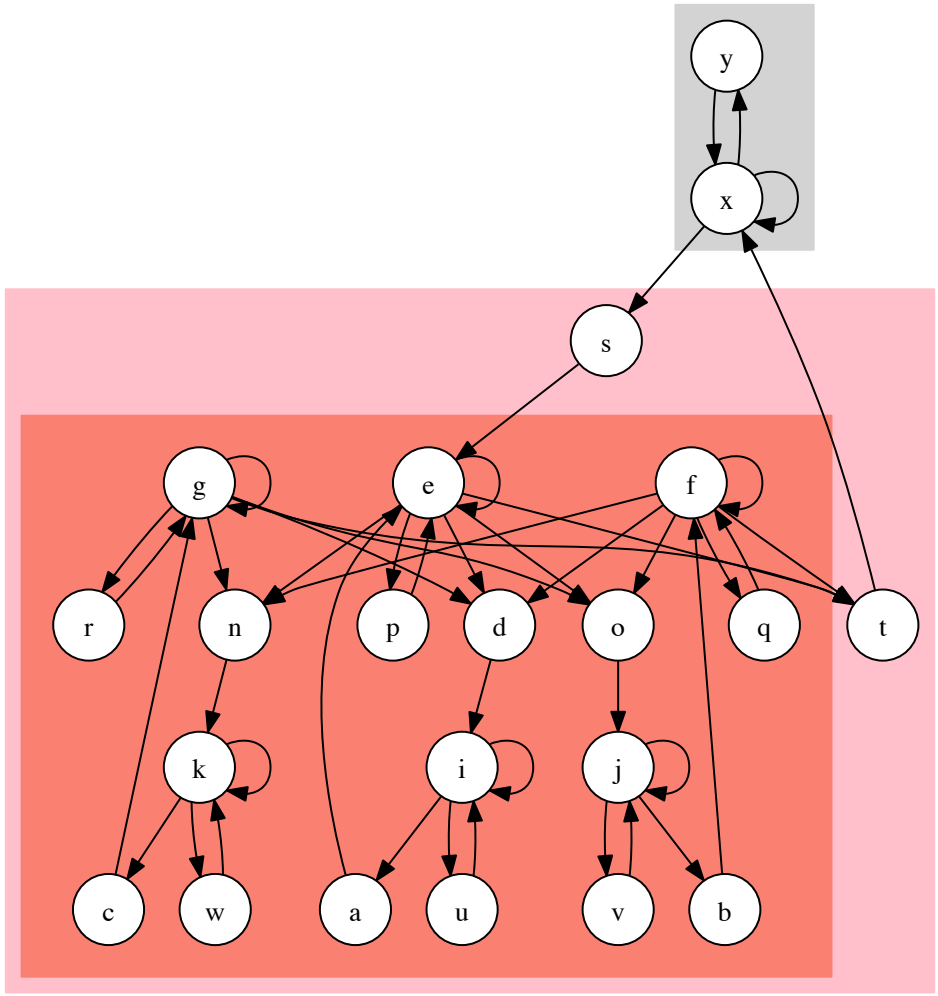
newL0,trans,strands



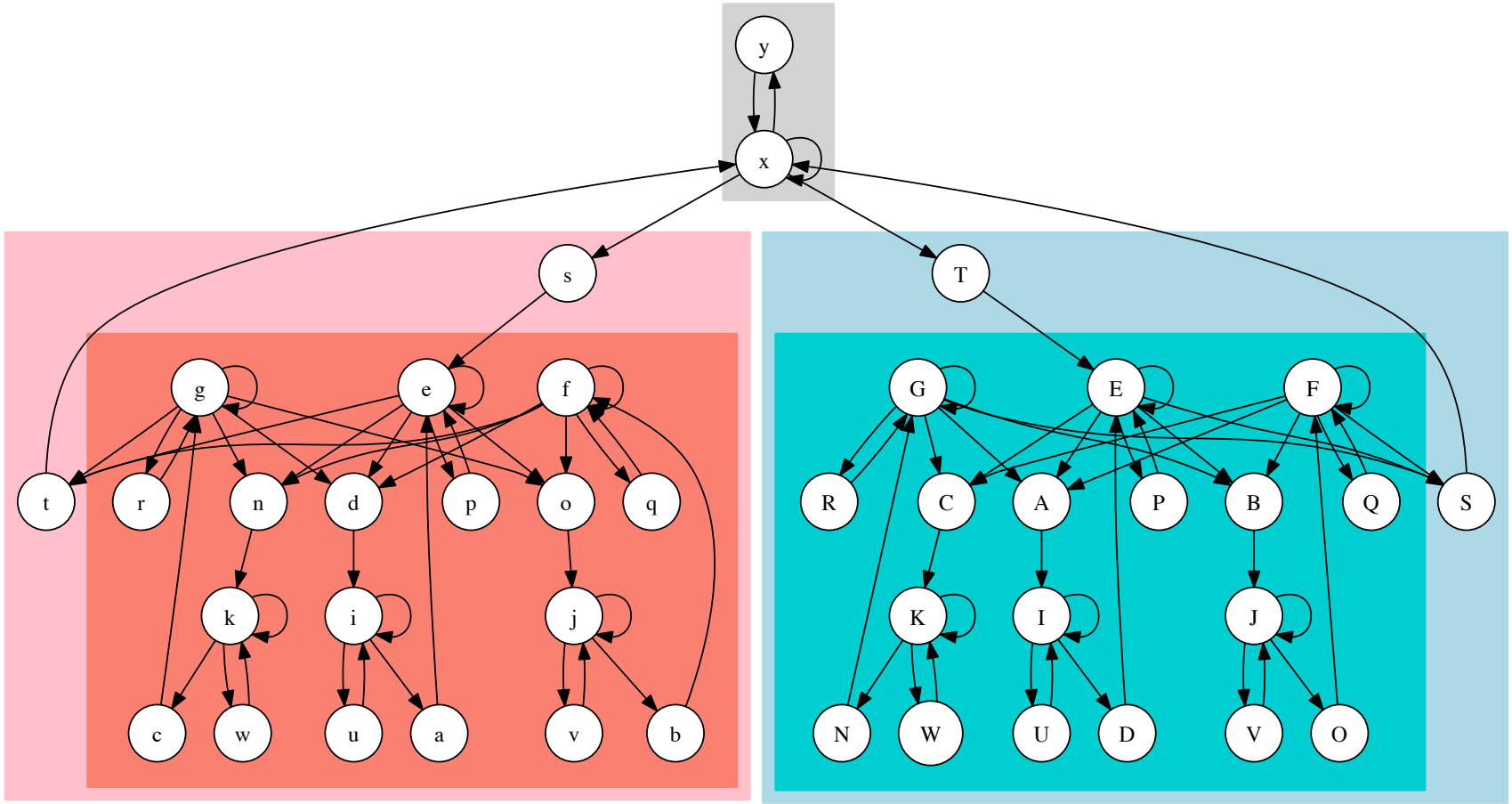
L1



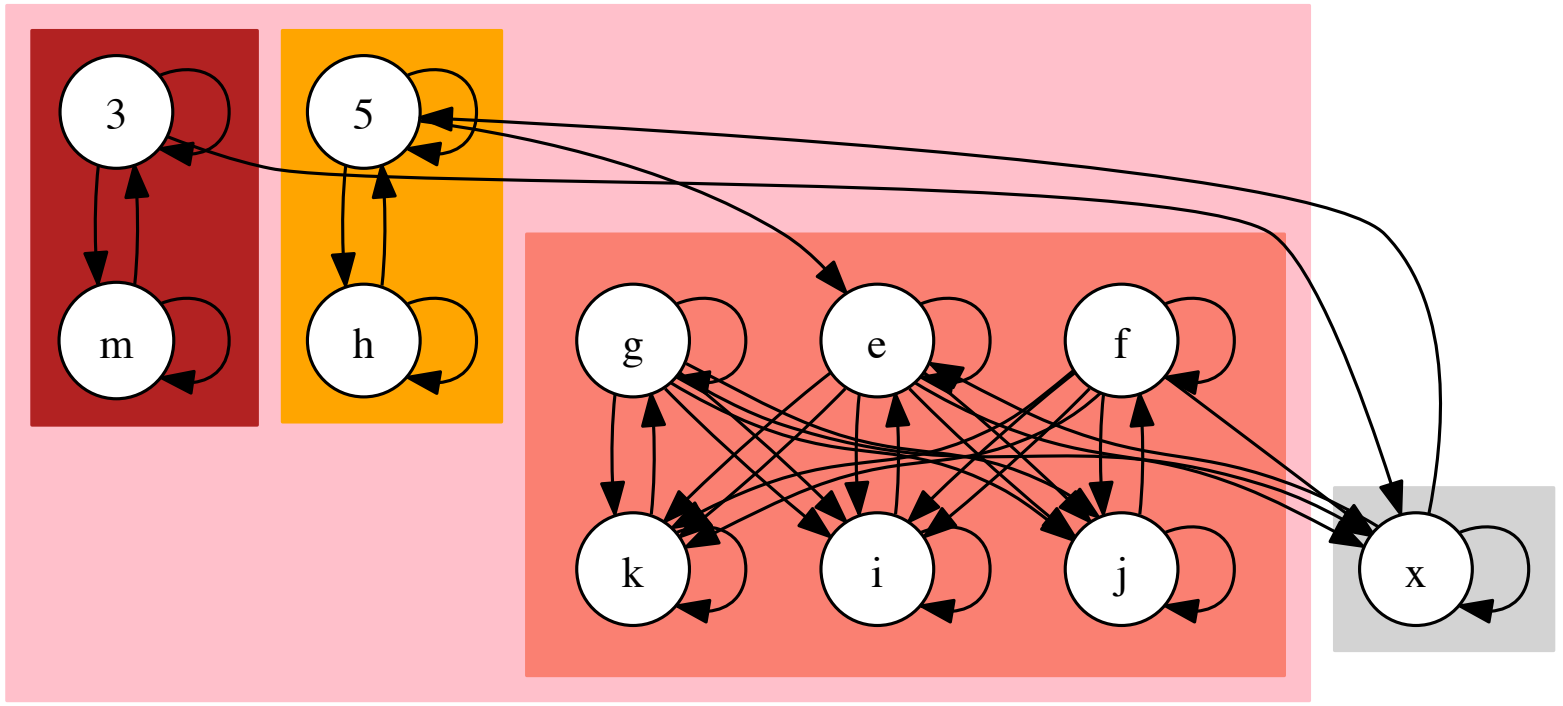
L1,trans



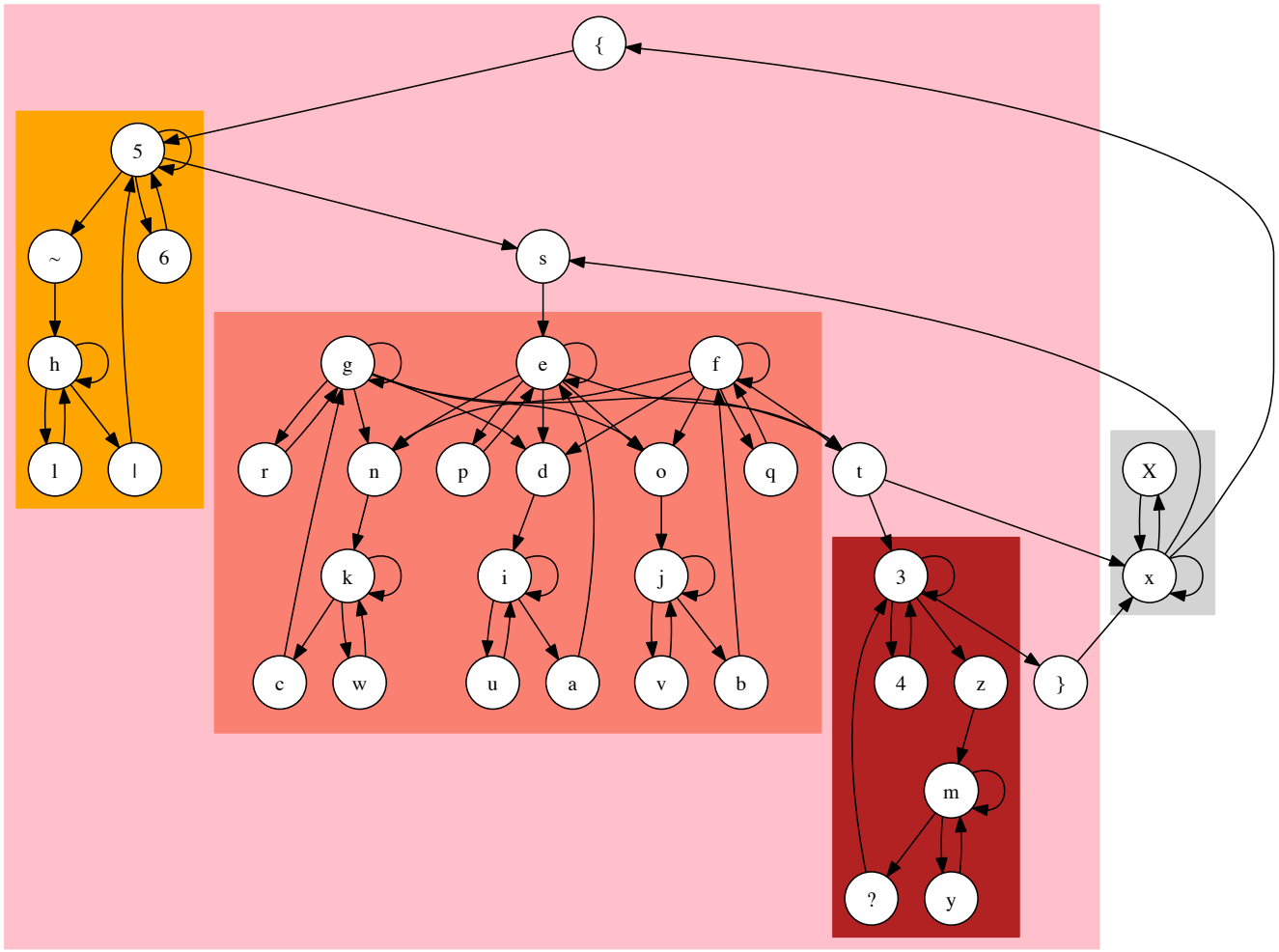
L1,trans,strands



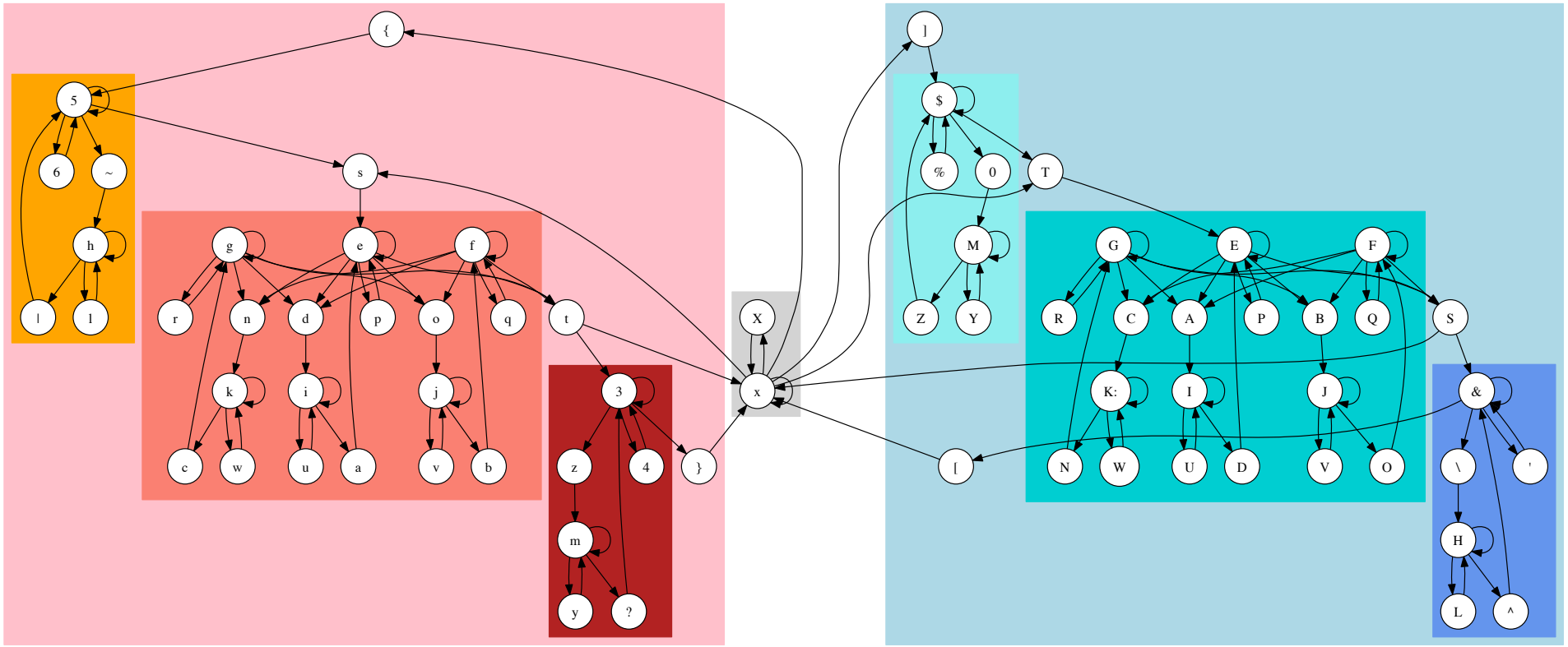
L1utr



L1utr,trans



L1utr,trans,strands



A full description of each letter follows:

x	intergenic
X	remain intergenic
{,[	TSS (forward,reverse)
5,&	5’ UTR exon (f,r)
6,’	remain 5’ UTR exon (f,r)
~,^	5’ UTR donor (f,r)
h,H	5’ UTR intron (f,r)
l,L	remain 5’ UTR intron (f,r)
|,\	5’ UTR acceptor (f,r)
s,S	start site (f,r)
e,E	exon starting with a full reading frame (f,r)
p,P	remain e(E) exon (f,r)
d,D	donor site after a full reading frame (f,r)
i,I	intron following a d(D) donor (f,r)
u,U	remain i(I) intron (f,r)
a,A	acceptor preceding a full reading frame (f,r)
f,F	exon starting with 2 letters of a reading frame (f,r)
q,Q	remain f(F) exon (f,r)
o,O	donor site after 1 letter of a reading frame (f,r)
j,J	intron following a o(O) donor (f,r)
v,V	remain j(J) intron (f,r)
b,B	acceptor preceeding 2 letters of a reading frame
g,G	exon starting with 1 letter of a reading frame (f,r)
r,R	remain g(G) exon (f,r)
n,N	donor site after 2 letters of a reading frame (f,r)
k,K	intron flowwing an n(N) donor (f,r)
w,W	remain k(K) intron (f,r)
c,C	acceptor preceding 1 letter of a reading frame
t,T	stop site (f,r)
3,$	3’ UTR exon (f,r)
4,%	remain 3’ UTR exon (f,r)
z,Z	3’ UTR donor (f,r)
m,M	3’ UTR intron (f,r)
y,Y	remain 3’ UTR intron (f,r)
?,0	3’ UTR acceptor (f,r)
},]	TTS (f,r)

Unused characters: ! # + , - . / : < = > @ _ 1 2 7 8 9