Multi-founder HIV sequence simulator

Description

Simulate sequences that recapitulate the site-by-site amino acid diversity of a multi-founder HIV-1 acute infection.

Usage

```
MFSIM(mat,seed,al,n,name,founder_seqs,training=1,tmat=F,indel=F,founder_max=T,write=F,dif_prop=F,prop)
```

Arguments

`mat`	a character matrix of an amino acid multiple sequence alignment that will be used to calculate a transition matrix; or transition matrix in table format if tmat=T.
`seed`	a character string of the amino acid sequence to seed the simulation.
`al`	a character matrix of an amino acid multiple sequence alignment of the training sequences.
`n`	the number of sequences to simulate under each founder model.
`name`	the name for the output file.
`founder_seqs`	a list of sequences included in each founder lineage; if training==2, each list should be a list of sequences included in each founder lineage for each participant (participants may have different numbers of founder lineages).
`training`	whether the training alignment is comprised of sequences from a single participant (1) or multiple participants (2).
`tmat`	whether mat is a transition matrix in table format (T) or an amino acid multiple sequence alignment that will be used to calculate a transition matrix (F).
`indel`	whether to include indels (T) in simulated sequences.
`founder_max`	whether the percentage of non-consensus residues used at each site in the training alignment is the maximum (T) or drawn randomly (F) across participants.
`write`	whether the simulated sequences should be written to a fasta file.

`dif_prop`
: whether specified proportions of sequences should be returned for each simulated alignment.

`prop`
: a list of the proportion of sequences to be returned for each simulated alignment.

Value

a list with the following components:

`seqs`
: a list of simulated sequences.

`names`
: the names of simulated sequences.

`ref_noncons`
: the percentage of non-consensus residues at each site in the training MSA

`sim_noncon`
: the percentage of non-consensus residues at each site in the simulated MSA

`sim_noncon_founder`
: a list of the the percentage of non-consensus residues at each site in the MSAs simulated from each training founder lineage.

`sim_founder_dists`
: a list of distance matrices calculated from the MSAs simulated from each training founder lineage.

`sim_founder_polies`
: a list of the the polymorphic sites calculated from the MSAs simulated from each training founder lineage.

`aa_sim`
: a MSA (AAbin) of all simulated sequences.

`aa_sim_founder`
: a list of MSAs (AAbin) of sequences simulated from each training founder lineage.

Author(s)

E Lewitus

References

Lewitus, E., Hoang, J., Li, Y., Bai, H., and Rolland, M. Optimal sequence-based design for multi-antigen HIV-1 vaccines using minimally distant antigens. PLoS Computional Biology.

Examples

```
## Simulate sequences
```

```
        #mat: HIV-1 subtype C alignment
        fasC<-bio3d::read.fasta('env_subtypeC_post2010_alignment.fas')$ali

        #seed: HIV-1 subtype C consensus sequence
        consensusC<-bio3d::consensus(fasC)$seq
        consensusC<-consensusC[which(consensusC!='-')]

        #al: multi-founder acute infection alignment
        mf_fas<-bio3d::read.fasta('RV217_multifounder_alignment.fas')$ali

        #number of sequences to simulate
        n=1e3

        #table delineating founder variants
        mf_tab<-read.csv('mf_founder_tab.csv')

## Simulate sequences with a training alignment from a single participant
        #al: one multi-founder acute infection alignment
        mf_fas_20337<-mf_fas[grep('20337',rownames(mf_fas)),]

        name='MF_sim_output_20337'

        #founder_seqs: sequence names belonging to each founder lineage
        mf_tab_20337<-mf_tab[,grep('20337',colnames(mf_tab))]
        founder_seqs_list_20337<-list(
        rownames(mf_fas_20337)[which(rownames(mf_fas_20337)%in%mf_tab_20337)],
        rownames(mf_fas_20337)[which(!rownames(mf_fas_20337)%in%mf_tab_20337)]
        )

        #run
        one_mfsim<-MFSIM(mat=fasC,seed=consensusC,al=mf_fas_20337,
                n=n,name=name,founder_seqs=founder_seqs_list_20337,
                training=1,write=T)

## Simulate sequences with a training alignment from more than one participant

        name='several_MF_sim_output'

        #proportion of sequences simulated on each founder training alignment
        prop<-c(1,0.5,0.75,1)

        #founder_seqs: sequence names belonging to each founder lineage
        mf_names<-c('10220','20337','30124','40363')
        founder_seqs_list<-lapply(mf_names,function(i){
                h<-mf_fas[grep(i,rownames(mf_fas)),]
                j<-mf_tab[,grep(i,colnames(mf_tab))]
                list(rownames(h)[which(rownames(h)%in%j)],
                rownames(h)[which(!rownames(h)%in%j)])
        })

        #run
        several_mfsim<-MFSIM(mat=fasC,seed=consensusC,al=mf_fas,
                n=n,name=name,founder_seqs=founder_seqs_list,
                training=2,indel=F,founder_max=F,write=T,dif_prop=T,prop=prop)
```