Table of Contents

```
clc
clear all

% Date: 18.04.2020
```

This process is with INDELible data with cut-off, m=20

Add paths

```
baseType0= 'n_4_a_0_5'; % % %
baseType1= 'n_4_a_1_0';
baseType2= 'n_4_a_2_0';
baseType3= 'n_4_a_3_0';

addpath("/Users/pouloeld/Documents/Statistics2/INDELible/True/m_20")
addpath(strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType0))
addpath(strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType1))
addpath(strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType2))
addpath(strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType3))
```

Readin TRUE MSA

```
%INDELible
True_I_Path="/Users/pouloeld/Documents/Statistics2/INDELible/True/
m_20";
True_I_files= dir(fullfile(True_I_Path,'*_TRUE_1.fasta'));
True_I_Names= {True_I_files.name};
tk_i= regexp(True_I_Names,'((?<=out_)\d*)|((?<=_TRUE_1))','match');
[~,ti]= sortrows(str2double(cat(1,tk_i{:})));
```

```
True_I_Names= True_I_Names(ti); %ti for INDELible tp for PIPJava tr
 for real
nIter_I= length(True_I_Names);
```

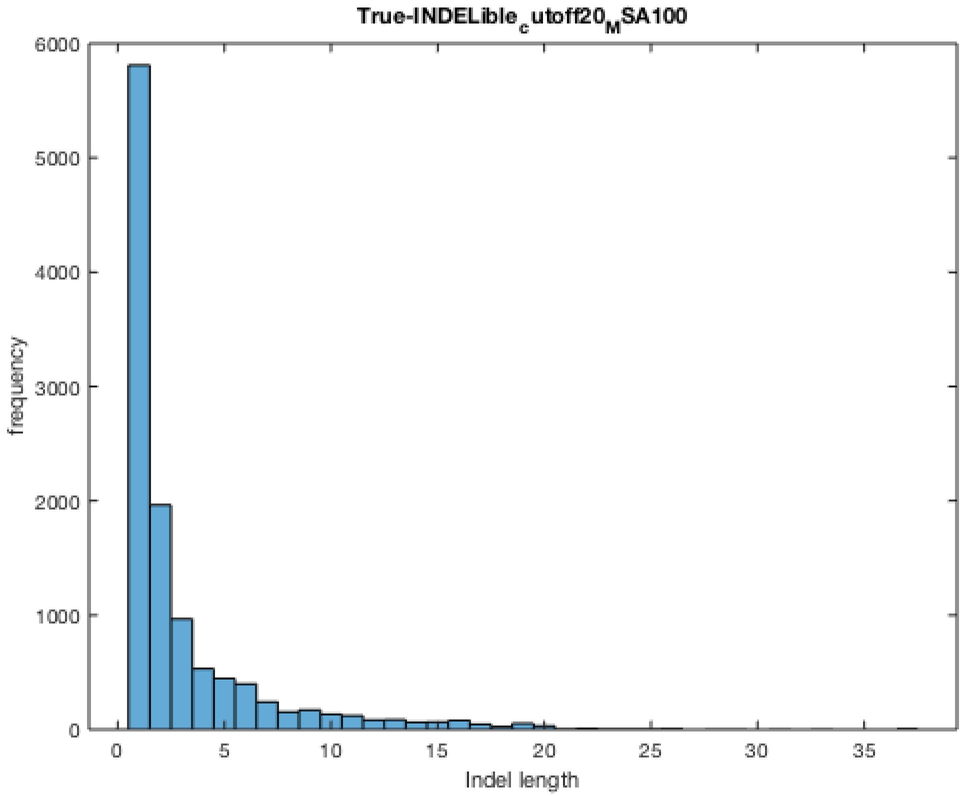
Readin all Inferred MSAs

```
% ProPIP + Gamma n=4, a=0.5
Inf_I_Path0= strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType0);
Inf_I_files0= dir(fullfile(Inf_I_Path0,'*.fasta'));
Inf_I_Names0= {Inf_I_files0.name};
tk_i0= regexp(Inf_I_Names0,'((?
<=ProPIPgamma0inferredMSA_I_)\d*)','match');
[~,ti0]= sortrows(str2double(cat(1,tk_i0{:})));
Inf_I_Names0= Inf_I_Names0(ti0); %tii for INDELible tpp for PIPJava
 trr for real
nIter_I0= length(Inf_I_Names0);

% ProPIP + Gamma n=4, a=1.0
Inf_I_Path1= strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType1);
Inf_I_files1= dir(fullfile(Inf_I_Path1,'*.fasta'));
Inf_I_Names1= {Inf_I_files1.name};
tk_i1= regexp(Inf_I_Names1,'((?
<=ProPIPgamma1inferredMSA_I_)\d*)','match');
[~,ti1]= sortrows(str2double(cat(1,tk_i1{:})));
Inf_I_Names1= Inf_I_Names1(ti1); %tii for INDELible tpp for PIPJava
 trr for real
nIter_I1= length(Inf_I_Names1);

% ProPIP + Gamma n=4, a=2.0
Inf_I_Path2= strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType2);
Inf_I_files2= dir(fullfile(Inf_I_Path2,'*.fasta'));
Inf_I_Names2= {Inf_I_files2.name};
tk_i2= regexp(Inf_I_Names2,'((?
<=ProPIPgamma2inferredMSA_I_)\d*)','match');
[~,ti2]= sortrows(str2double(cat(1,tk_i2{:})));
Inf_I_Names2= Inf_I_Names2(ti2); %tii for INDELible tpp for PIPJava
 trr for real
nIter_I2= length(Inf_I_Names2);

% ProPIP + Gamma n=4, a=3.0
Inf_I_Path3= strcat('/Users/pouloeld/Documents/Statistics2/INDELible/
ProPIP/Gamma','/',baseType3);
Inf_I_files3= dir(fullfile(Inf_I_Path3,'*.fasta'));
Inf_I_Names3= {Inf_I_files3.name};
tk_i3= regexp(Inf_I_Names3,'((?
<=ProPIPgamma3inferredMSA_I_)\d*)','match');
[~,ti3]= sortrows(str2double(cat(1,tk_i3{:})));
Inf_I_Names3= Inf_I_Names3(ti3); %tii for INDELible tpp for PIPJava
 trr for real
nIter_I3= length(Inf_I_Names3);
```

INDEL length per True_MSA

INDELible

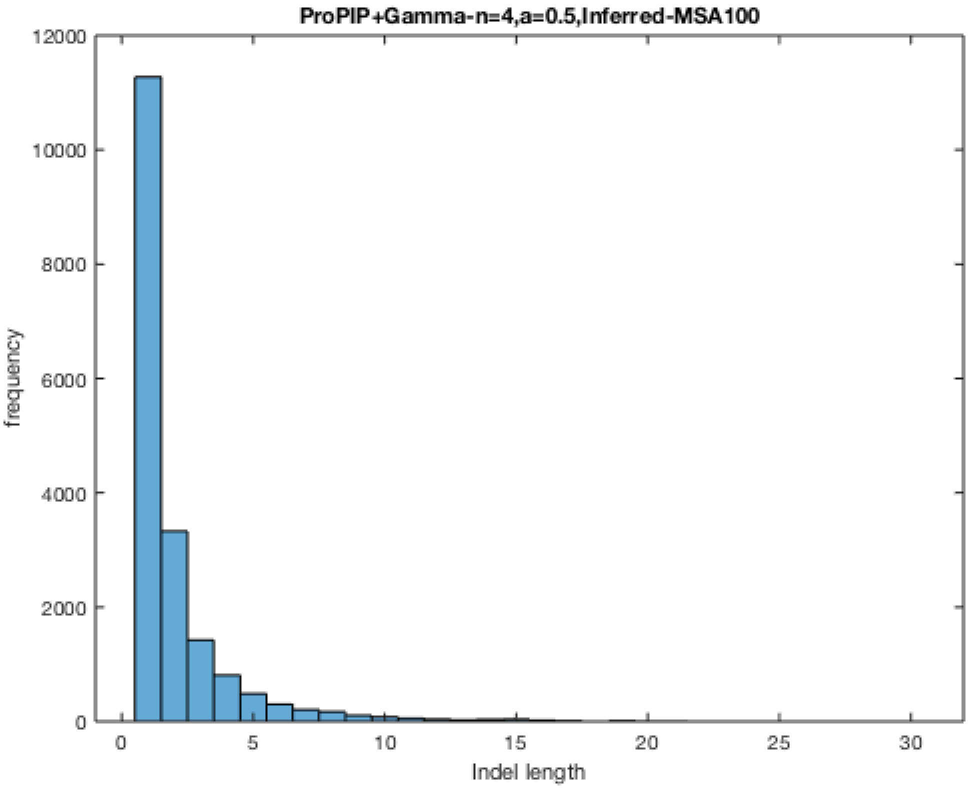
```
count=0;
True_I_count= [];
for namesI=1:nIter_I0
    True_I_MSA= fastaread(True_I_Names(namesI));
    True_I_countMSA= [];
    for nTaxa=1:8
        for seqL=1:length(True_I_MSA(nTaxa).Sequence)
            if True_I_MSA(nTaxa).Sequence(seqL) == '-'
                count= count+1;
            elseif True_I_MSA(nTaxa).Sequence(seqL) ~= '-'
                if count ~= 0
                    True_I_count= [True_I_count,count];
                    True_I_countMSA= [True_I_countMSA,count];
                    count=0;
                end
            end
        end
        if count ~=0
            True_I_count= [True_I_count,count];
            True_I_countMSA= [True_I_countMSA,count];
            count=0;
        end
    end
%     figure,
%     histogram(True_I_countMSA,'Normalization','count')
%     title(strcat('True-INDELible_cutoff20_MSA',int2str(namesI)))
%     xlabel('Indel length')
%     ylabel('frequency')

    %MsaL_I= seqL;
end

    figure,
    histogram(True_I_count,'Normalization','count')
    title(strcat('True-INDELible_cutoff20_MSA',int2str(namesI)))
    xlabel('Indel length')
    ylabel('frequency')
```



INDEL length per Inf_MSA

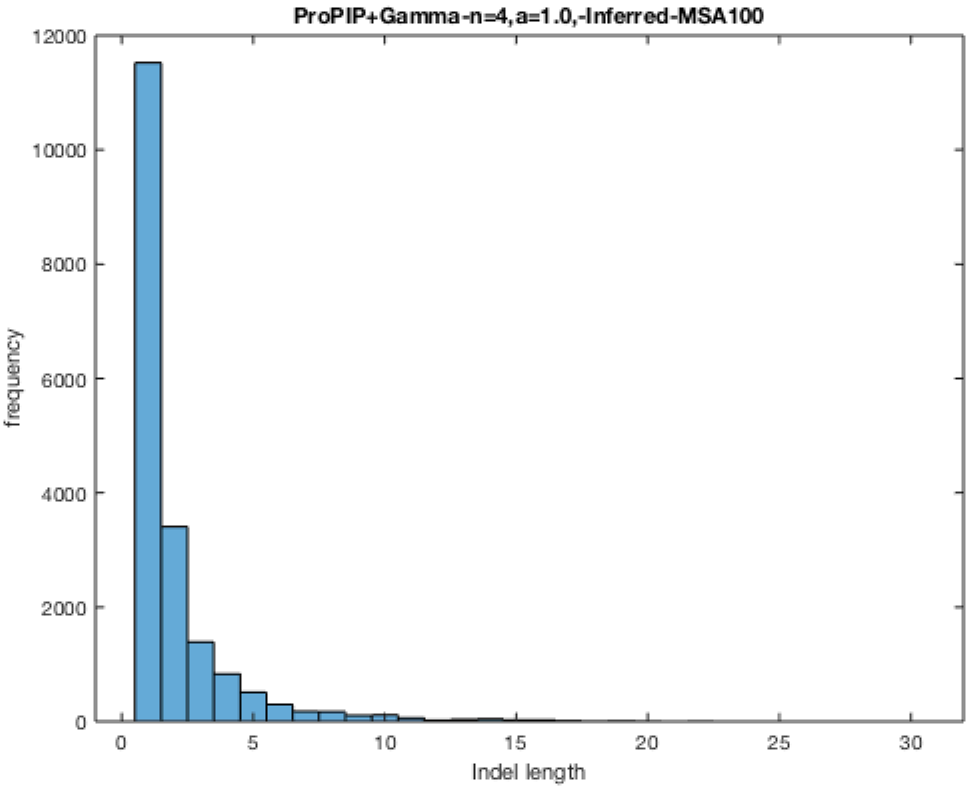
ProPIP + Gamma n=4, a=0.5

```
count=0;
Inf_I_count0= [];
maxMSA0=0;
for namesI0=1:nIter_I0
    Inf_I_MSA0= fastaread(Inf_I_Names0(namesI0));
    Inf_I_countMSA0= [];
    for nTaxa=1:8
        for seqL=1:length(Inf_I_MSA0(nTaxa).Sequence)
            if Inf_I_MSA0(nTaxa).Sequence(seqL) == '-'
                count= count+1;
            elseif Inf_I_MSA0(nTaxa).Sequence(seqL) ~= '-'
                if count ~= 0
                    Inf_I_count0= [Inf_I_count0,count];
                    Inf_I_countMSA0= [Inf_I_countMSA0,count];
                    if count>29
                        maxMSA0=namesI0;
                    end
                    count=0;
                end
            end
        end
        if count ~=0
            Inf_I_count0= [Inf_I_count0,count];
            Inf_I_countMSA0= [Inf_I_countMSA0,count];
            if count>29
                maxMSA0=namesI0;
            end
            count=0;
        end
    end

%     if namesI3==41
%         MSA_stats{1}=
 [length(Inf_I_countMSA0),max(Inf_I_countMSA0),mean(Inf_I_countMSA0),median(Inf_I_
%     end
%
%     disp(namesI0)
%     disp(Inf_I_countMSA0)

%     figure,
%     histogram(Inf_I_countMSA0,'Normalization','count')
%     title(strcat('ProPIP+Gamma-n=4,a=0.5,Inferred-
MSA',int2str(namesI0)))
%     xlabel('Indel length')
%     ylabel('frequency')



    %MsaL_I0= seqL;

end

    figure,
    histogram(Inf_I_count0,'Normalization','count')
    title(strcat('ProPIP+Gamma-n=4,a=0.5,Inferred-
MSA',int2str(namesI0)))
    xlabel('Indel length')
    ylabel('frequency')
```



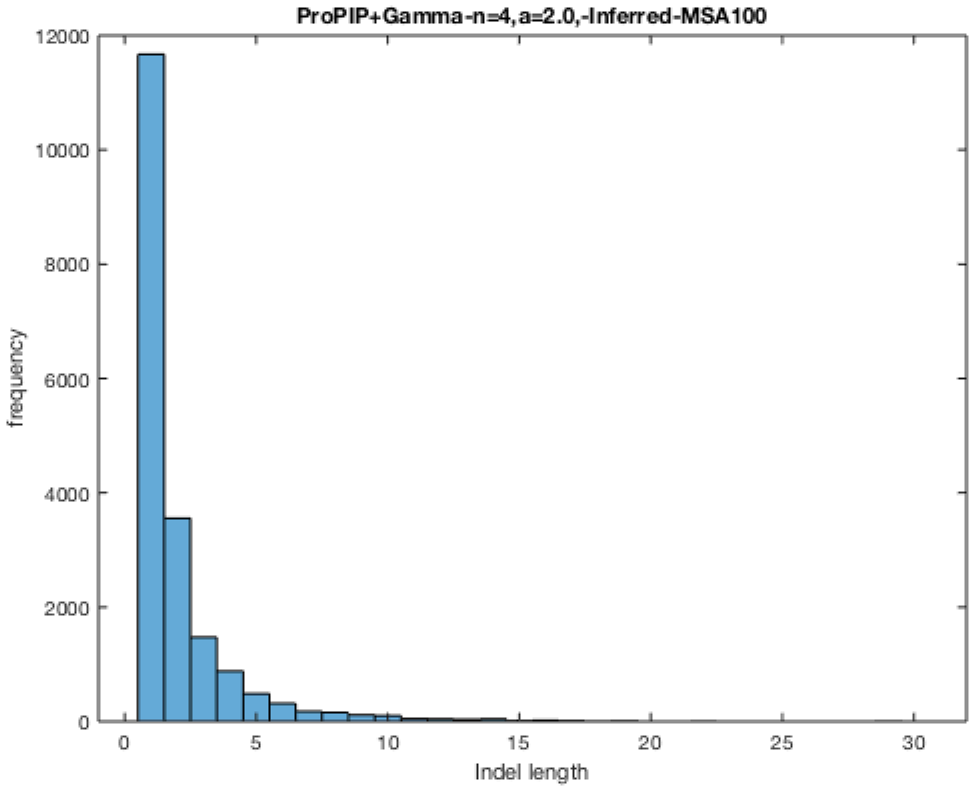
ProPIP + Gamma n=4, a=1.0

```
count=0;
Inf_I_count1= [];
maxMSA1=0;
for namesI1=1:nIter_I1
    Inf_I_MSA1= fastaread(Inf_I_Names1(namesI1));
    Inf_I_countMSA1= [];
    for nTaxa=1:8
        for seqL=1:length(Inf_I_MSA1(nTaxa).Sequence)
            if Inf_I_MSA1(nTaxa).Sequence(seqL) == '-'
                count= count+1;
            elseif Inf_I_MSA1(nTaxa).Sequence(seqL) ~= '-'
                if count ~= 0
                    Inf_I_count1= [Inf_I_count1,count];
                    Inf_I_countMSA1= [Inf_I_countMSA1,count];
                    if count>29
                        maxMSA1=namesI1;
                    end
                    count=0;
                end
            end
        end
        if count ~=0
            Inf_I_count1= [Inf_I_count1,count];
            Inf_I_countMSA1= [Inf_I_countMSA1,count];
            if count>29
                maxMSA1=namesI1;
            end
            count=0;
        end
    end

%     if namesI1==41
%         MSA_stats{1}=
 [length(Inf_I_countMSA1),max(Inf_I_countMSA1),mean(Inf_I_countMSA1),median(Inf_I_
%     end
%
%     disp(namesI1)
%     disp(Inf_I_countMSA1)
%
%     figure,
%     histogram(Inf_I_countMSA1,'Normalization','count')
%     title(strcat('ProPIP+Gamma-n=4,a=1.0,-Inferred-
MSA',int2str(namesI1)))
%     xlabel('Indel length')
%     ylabel('frequency')



    %MsaL_I1= seqL;

end


    figure,
    histogram(Inf_I_count1,'Normalization','count')
    title(strcat('ProPIP+Gamma-n=4,a=1.0,-Inferred-
MSA',int2str(namesI1)))
    xlabel('Indel length')
    ylabel('frequency')
```



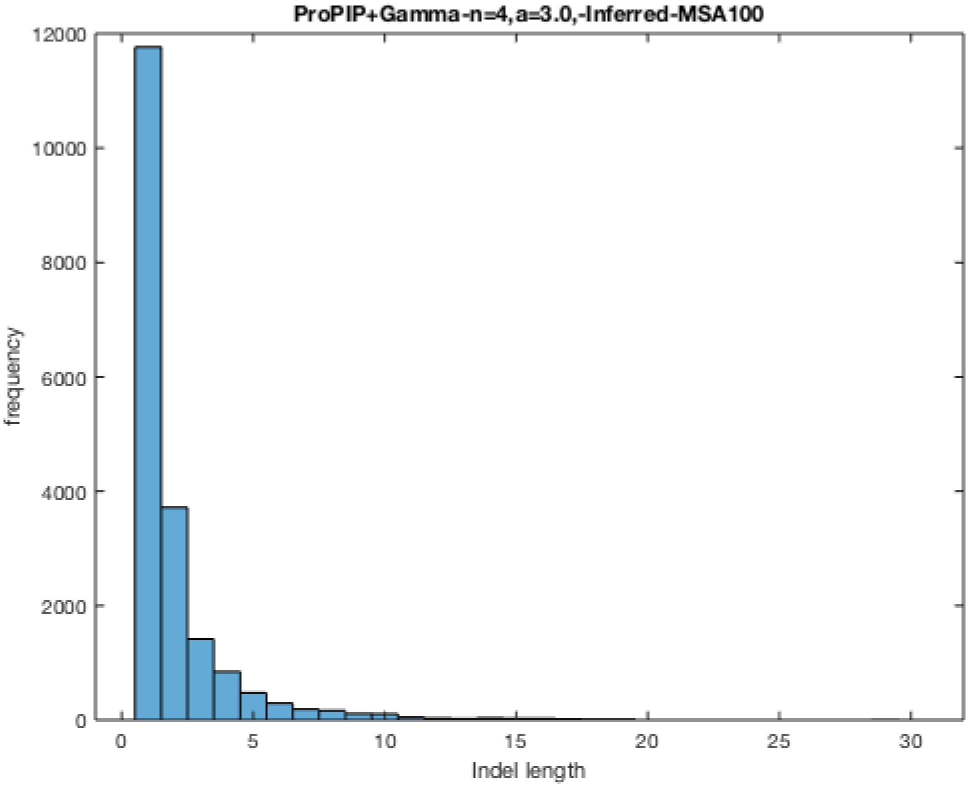
ProPIP + Gamma n=4, a=2.0

```
count=0;
Inf_I_count2= [];
maxMSA2=0;
for namesI2=1:nIter_I2
    Inf_I_MSA2= fastaread(Inf_I_Names2(namesI2));
    Inf_I_countMSA2= [];
    for nTaxa=1:8
        for seqL=1:length(Inf_I_MSA2(nTaxa).Sequence)
            if Inf_I_MSA2(nTaxa).Sequence(seqL) == '-'
                count= count+1;
            elseif Inf_I_MSA2(nTaxa).Sequence(seqL) ~= '-'
                if count ~= 0
                    Inf_I_count2= [Inf_I_count2,count];
                    Inf_I_countMSA2= [Inf_I_countMSA2,count];
                    if count>29
                        maxMSA2=namesI2;
                    end
                    count=0;
                end
            end
        end
        if count ~=0
            Inf_I_count2= [Inf_I_count2,count];
            Inf_I_countMSA2= [Inf_I_countMSA2,count];
            if count>29
                maxMSA2=namesI2;
            end
            count=0;
        end
    end

%     if namesI2==41
%         MSA_stats{1}=
 [length(Inf_I_countMSA2),max(Inf_I_countMSA2),mean(Inf_I_countMSA2),median(Inf_I_
%     end
%
%     disp(namesI2)
%     disp(Inf_I_countMSA2)
%
%     figure,
%     histogram(Inf_I_countMSA2,'Normalization','count')
%     title(strcat('ProPIP+Gamma-n=4,a=2.0,-Inferred-
MSA',int2str(namesI2)))
%     xlabel('Indel length')
%     ylabel('frequency')


    %MsaL_I2= seqL;

end


    figure,
    histogram(Inf_I_count2,'Normalization','count')
    title(strcat('ProPIP+Gamma-n=4,a=2.0,-Inferred-
MSA',int2str(namesI2)))
    xlabel('Indel length')
    ylabel('frequency')
```



ProPIP + Gamma n=4, a=3.0

```
count=0;
Inf_I_count3= [];
maxMSA3=0;
for namesI3=1:nIter_I3
    Inf_I_MSA3= fastaread(Inf_I_Names3(namesI3));
    Inf_I_countMSA3= [];
    for nTaxa=1:8
        for seqL=1:length(Inf_I_MSA3(nTaxa).Sequence)
            if Inf_I_MSA3(nTaxa).Sequence(seqL) == '-'
                count= count+1;
            elseif Inf_I_MSA3(nTaxa).Sequence(seqL) ~= '-'
                if count ~= 0
                    Inf_I_count3= [Inf_I_count3,count];
                    Inf_I_countMSA3= [Inf_I_countMSA3,count];
                    if count>29
                        maxMSA3=namesI3;
                    end
                    count=0;
                end
            end
        end
        if count ~=0
            Inf_I_count3= [Inf_I_count3,count];
            Inf_I_countMSA3= [Inf_I_countMSA3,count];
            if count>29
                maxMSA3=namesI3;
            end
            count=0;
        end
    end

%     if namesI3==41
%         MSA_stats{1}=
 [length(Inf_I_countMSA3),max(Inf_I_countMSA3),mean(Inf_I_countMSA3),median(Inf_I_
%     end
%
%     disp(namesI3)
%     disp(Inf_I_countMSA3)
%
%     figure,
%     histogram(Inf_I_countMSA3,'Normalization','count')
%     title(strcat('ProPIP+Gamma-n=4,a=3.0,-Inferred-
MSA',int2str(namesI3)))
%     xlabel('Indel length')
%     ylabel('frequency')


    %MsaL_I3= seqL;

end

    figure,
    histogram(Inf_I_count3,'Normalization','count')
    title(strcat('ProPIP+Gamma-n=4,a=3.0,-Inferred-
MSA',int2str(namesI3)))
    xlabel('Indel length')
    ylabel('frequency')

stat{1}=
 [length(True_I_count),max(True_I_count),mean(True_I_count),median(True_I_count),s
stat0{1}=
 [length(Inf_I_count0),max(Inf_I_count0),mean(Inf_I_count0),median(Inf_I_count0),s
stat1{1}=
 [length(Inf_I_count1),max(Inf_I_count1),mean(Inf_I_count1),median(Inf_I_count1),s
stat2{1}=
 [length(Inf_I_count2),max(Inf_I_count2),mean(Inf_I_count2),median(Inf_I_count2),s
stat3{1}=
 [length(Inf_I_count3),max(Inf_I_count3),mean(Inf_I_count3),median(Inf_I_count3),s
```



Published with MATLAB® R2019b