

```

1
2
3     typedef double XFLOAT
4     typedef double OTA_FLOAT
5
6 using std
7
8
9 {
10
11 void* PreAlignment_Init(XFLOAT const* refBuff, long numRefSamples,
12                        XFLOAT const* degBuff, long numDegSamples,
13                        int samplingFreq, int appType, FILE* pLogFile, void* mh)
14 {
15
16     int iRetVal          = SQERR_UNKNOWN_ERROR
17     PreAlignment *Algo = NULL
18     OPTTRY
19     {
20         OPTTRY
21         {
22
23             //Load input files and desired frame configuration
24             XFLOAT frameLengthInSec = (XFLOAT)0.032
25             XFLOAT frameStepInSec   = (XFLOAT)0.016
26             Algo = new PreAlignment(refBuff, numRefSamples, degBuff, numDegSamples,
27                                   appType, samplingFreq, 0, pLogFile, mh)
28
29             return (void*)Algo
30         }
31         catch (access_violation const &e)
32         {
33             stringstream errStream
34             errStream << e.what() << " at " << std::hex << e.where() << ": Bad " << (e.isWrite() ?
"write" : "read")
35             << " on " << e.badAddress() << std::endl
36             throw SQError(SQERR_OTHER, errStream.str())
37         }
38         catch (win32_exception const &e)
39         {
40             stringstream errStream
41             errStream << e.what() << " (code " << std::hex << e.code() << ") at " << e.where() <<
std::endl
42             throw SQError(SQERR_OTHER, errStream.str())
43         }
44     }
45     OPTCATCH ((string errorMsg))
46     {
47
48         delete Algo
49         Algo = NULL
50         return (void*)Algo
51     }
52     OPTCATCH ((SQError err))
53     {
54
55         delete Algo
56         Algo = NULL
57         return (void*)Algo
58     }
59     OPTCATCH ((...))
60     {
61
62         delete Algo
63         Algo = NULL
64         return (void*)Algo
65     }
66 }

```

```

67
68 XFLOAT PreAlignment_GetResamplingFac(void *PAHandle)
69 {
70     if (PAHandle == NULL)
71         return (XFLOAT)-1.0
72
73     PreAlignment *Algo = (PreAlignment*)PAHandle
74     XFLOAT fac = (XFLOAT)-1.0
75
76     OPTTRY
77     {
78         fac = Algo->GetResamplingFac()
79     }
80     OPTCATCH ((...))
81     {
82         fac = (XFLOAT)-1.0
83     }
84
85     return fac
86 }
87
88 XFLOAT PreAlignment_GetDegSNR(void *PAHandle)
89 {
90     if (PAHandle == NULL)
91         return (XFLOAT)-1.0
92
93     PreAlignment *Algo = (PreAlignment*)PAHandle
94     XFLOAT snr = (XFLOAT)-1.0
95
96     OPTTRY
97     {
98         snr = Algo->GetDegSNR()
99     }
100    OPTCATCH ((...))
101    {
102        snr = (XFLOAT)-1.0
103    }
104    return snr
105 }
106
107 XFLOAT PreAlignment_GetMatchQuality(void *PAHandle)
108 {
109     if (PAHandle == NULL)
110         return (XFLOAT)-1.0
111
112     PreAlignment *Algo = (PreAlignment*)PAHandle
113     XFLOAT matchQual = (XFLOAT)-1.0
114
115     OPTTRY
116     {
117         matchQual = Algo->GetMatchQuality()
118     }
119     OPTCATCH ((...))
120     {
121         matchQual = (XFLOAT)-1.0
122     }
123     return matchQual
124 }
125
126 bool PreAlignment_ExtremeMatchFound(void *PAHandle)
127 {
128     if (PAHandle == NULL)
129         return false
130
131     PreAlignment *Algo = (PreAlignment*)PAHandle
132     bool extremeMatchFound = false
133
134     OPTTRY

```

```

135     {
136         extremeMatchFound = Algo->ExtremeMatchFound()
137     }
138     OPTCATCH ((...))
139     {
140         extremeMatchFound = false
141     }
142     return extremeMatchFound
143 }
144
145 TA_SegList const* PreAlignment_GetSegList(void *PAHandle)
146 {
147     if (PAHandle == NULL)
148         return NULL
149
150     PreAlignment *Algo = (PreAlignment*)PAHandle
151     TA_SegList const *pResult = NULL
152     OPTTRY
153     {
154         pResult = Algo->GetMergedSegList()
155     }
156     OPTCATCH( (...))
157     {
158         pResult = NULL
159     }
160     return pResult
161 }
162
163 TraversalVecType const* PreAlignment_Traverse(void *PAHandle, XFLOAT frameLengthInSec, XFLOAT
frameStepInSec)
164 {
165     if (PAHandle == NULL)
166         return NULL
167
168     PreAlignment *Algo = (PreAlignment*)PAHandle
169     TraversalVecType const *pResult = NULL
170     OPTTRY
171     {
172         Algo->TraverseSegList(frameLengthInSec, frameStepInSec)
173         pResult = &(Algo->TraversalVec())
174     }
175     OPTCATCH ((...))
176     {
177         pResult = NULL
178     }
179     return pResult
180 }
181
182 int PreAlignment_Free(void **pPAHandle)
183 {
184     if (pPAHandle == NULL || *pPAHandle == NULL)
185         return -1
186
187     PreAlignment *Algo = (PreAlignment*)(*pPAHandle)
188     delete Algo
189     Algo = (PreAlignment*)NULL
190     *pPAHandle = NULL
191
192     return 0
193 }
194
195 }
196

```