NAME
read_reading, fread_reading – Read a trace file into a **Read** structure.

SYNOPSIS
#include <Read.h>

Read *read_reading(
char *filename,
int format);

Read *fread_reading(
FILE *fp,
char *filename,
int format);

DESCRIPTION
These functions read trace files into a **Read** structure. A variety of formats are supported including ABI, ALF and SCF. (Note that the first two are only supported when the library is used as part of the Staden Package.) Additionally, support for reading the plain (old) staden format files and Experiment files is included. Compressed trace files may also be read. Decompression is performed using either **gzip -d** or **uncompress** and is written to a temporary file for further processing. The temporary file is then read and removed.

When reading an experiment file the trace file referenced by the LN and LT line types is read. The QL, QR (left and right quality clips), SL and SR (left and right vector clips) are taken from the Experiment file to produce the cutoff information held within the **Read** structure. The **orig_trace** field of the **Read** structure will then contain the pointer to the experiment file structure and the **orig_trace_format** field will be set to **TT_EXP**.

The functions allocate a **Read** structure which is returned. To deallocate this structure use the **read_deallocate()** function.

read_reading() reads a trace from the specified filename and format. Formats available are **TT_SCF**, **TT_ABI**, **TT_ALF**, **TT_PLN**, **TT_EXP**and **TT_ANY**. Specifying format **TT_ANY** will attempt to automatically detect the corret format type by analysing the trace file for magic numbers and composition. The **format** field of the structure can then be used to determine the real trace type.

fread_reading() reads a trace from the specified file pointer. The filename argument is used for setting the **trace_name** field of the resulting structure, and for error messages. Otherwise the function is identical to the **read_reading()** function.

The **Read** structure itself is as follows.
typedef uint_2 TRACE;   /* for trace heights */

typedef struct
{
** int   format;   /* Trace file format */**
** char   *trace_name; /* Trace file name   */**

** int   NPoints;   /* No. of points of data */**
** int   NBases;   /* No. of bases   */**

** /* Traces */**
** TRACE   *traceA;   /* Array of length 'NPoints' */**
** TRACE   *traceC;   /* Array of length 'NPoints' */**
** TRACE   *traceG;   /* Array of length 'NPoints' */**
** TRACE   *traceT;   /* Array of length 'NPoints' */**
** TRACE   maxTraceVal; /* The maximal value in any trace */**

** /* Bases */**
** char   *base;   /* Array of length 'NBases' */**

```
        uint_2   *basePos;   /* Array of length 'NBases' */

        /* Cutoffs */
        int      leftCutoff;  /* Number of unwanted bases */
        int      rightCutoff; /* Number of unwanted bases */

        /* Miscellaneous Sequence Information */
        char     *info;       /* misc seq info, eg comments */

        /* Probability information */
        char     *prob_A;     /* Array of length 'NBases' */
        char     *prob_C;     /* Array of length 'NBases' */
        char     *prob_G;     /* Array of length 'NBases' */
        char     *prob_T;     /* Array of length 'NBases' */

        /* The original input format data, or NULL if inapplicable */
        int orig_trace_format;
        void *orig_trace;

    } Read;
```

RETURN VALUES

On successful completion, the **read_reading()** and **fread_reading()** functions return a pointer to a **Read** structure. Otherwise these functions return NULLRead (which is a null pointer).

SEE ALSO

write_reading(3), **fwrite_reading**(3), **deallocate_reading**(3), **scf**(4), **ExperimentFile**(4)