

JAFROC Development Kit User Guide

Hong-Jun Yoon, MSEE
Dev P Chakraborty, PhD

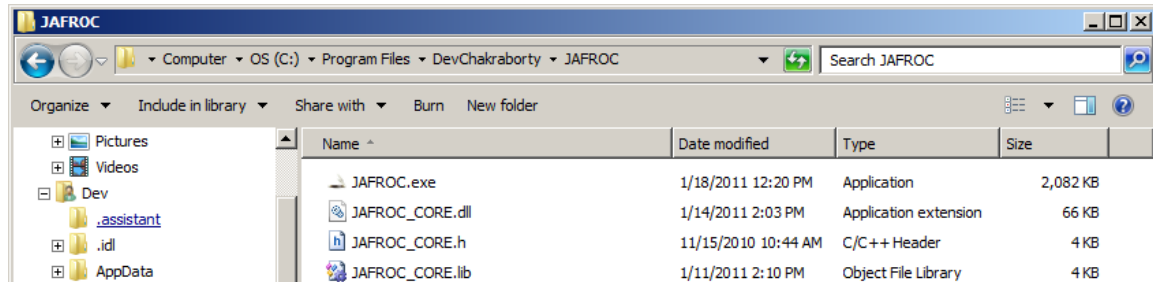
Windows XP, Vista and Windows 7 integrated software to perform custom analysis of FROC data

©Dev Chakraborty
www.devchakraborty.com

This will show you how to write your own JAFROC analysis program using the JAFROC Development Kit.

Preparation

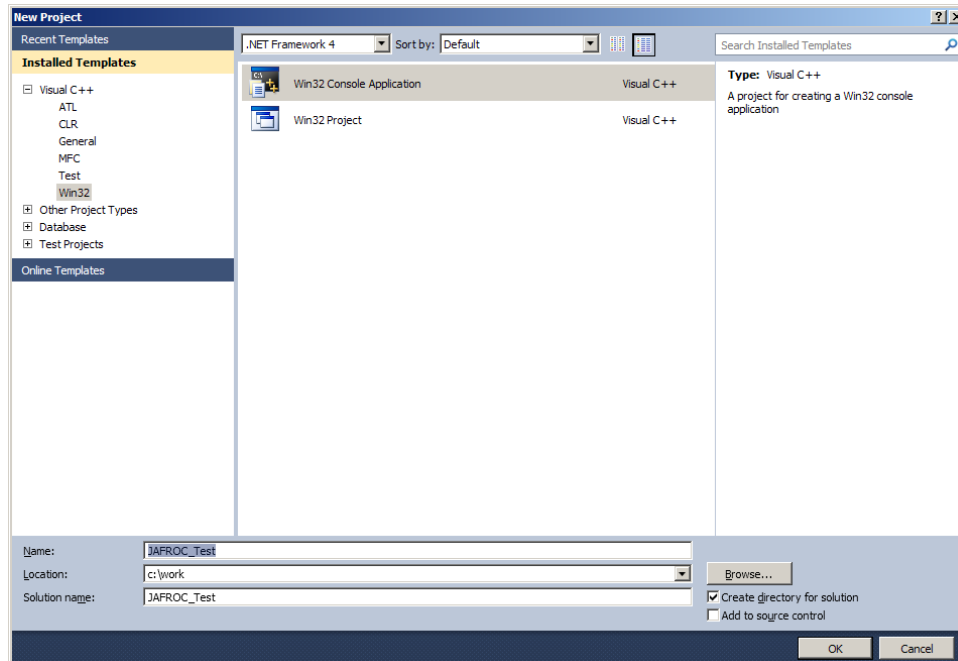
Files needed to develop a JAFROC analysis program are JAFROC_CORE.dll, JAFROC_CORE.lib, and JAFROC_CORE.h. These files are located in "C:\Program Files\DevChakraborty\JAFROC" folder.



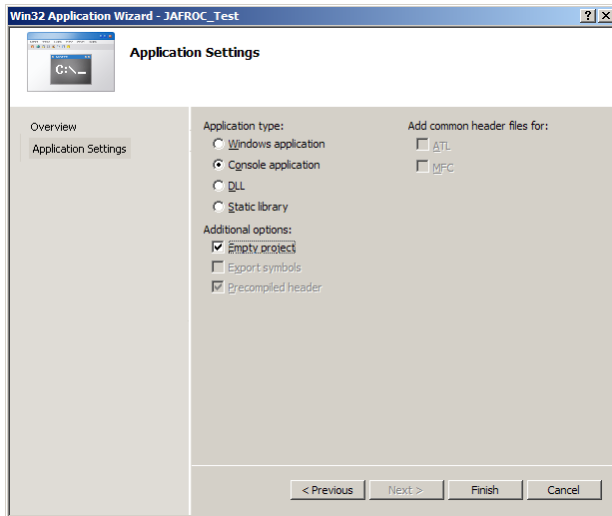
Create a new project

Any Win32 based C++ compiler can be used. This guide is based upon Microsoft Visual C++ 2010.

Select "File → New → Project" menu. In "Installed Templates" section, choose "Visual C++" → "Win32". Select "Win32 Console Application". Assign a name "JAFROC_Test".

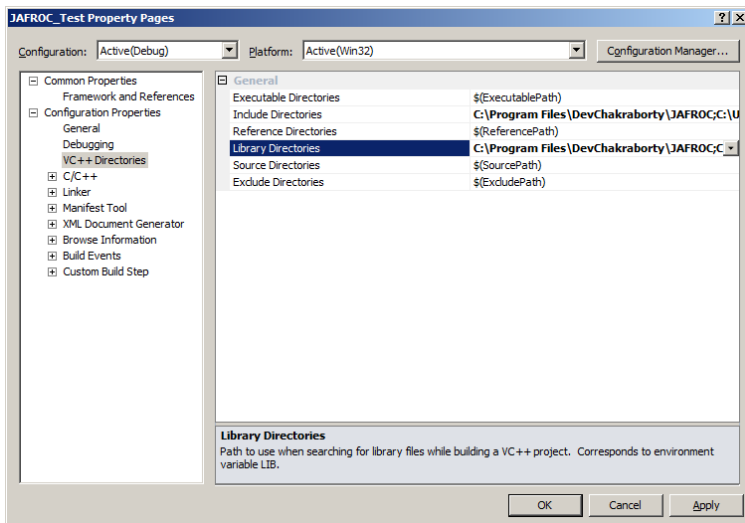


In "Application Settings" section, check "Empty Project". Press [Finish] to create a new project.

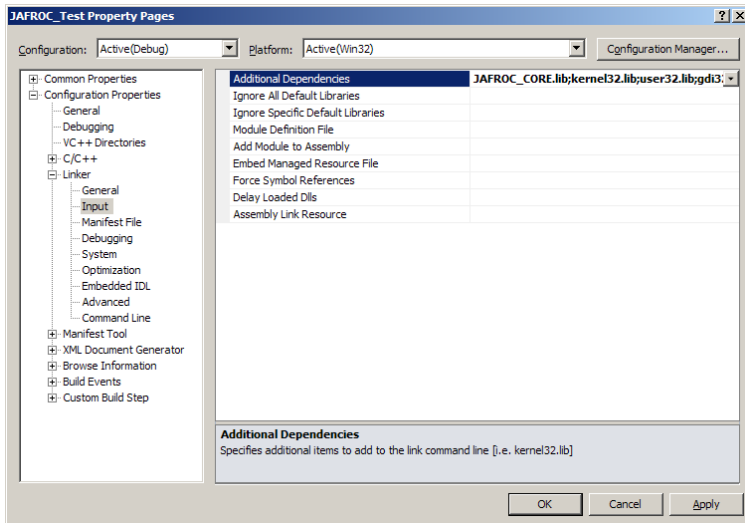


Directory Setting

Select "Project → Properties". In "Configuration Properties → VC++ Directories" section, add "C:\Program Files\DevChakraborty\JAFROC" folder path for "include" and "library" directories.

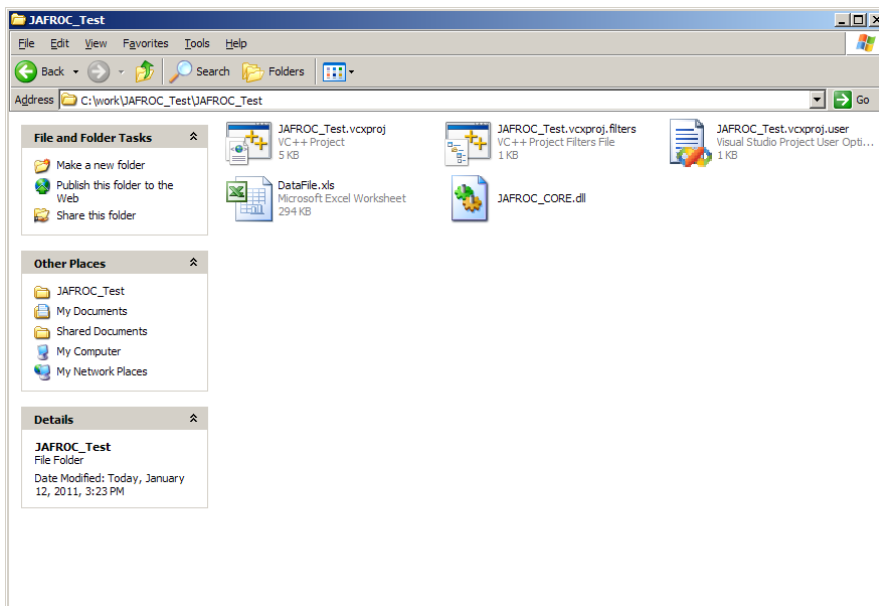


In "Configuration Properties → Linker → Input" section, add JAFROC_CORE.lib to the "Additional Dependencies" entry. Make sure to add semicolon between library file names.

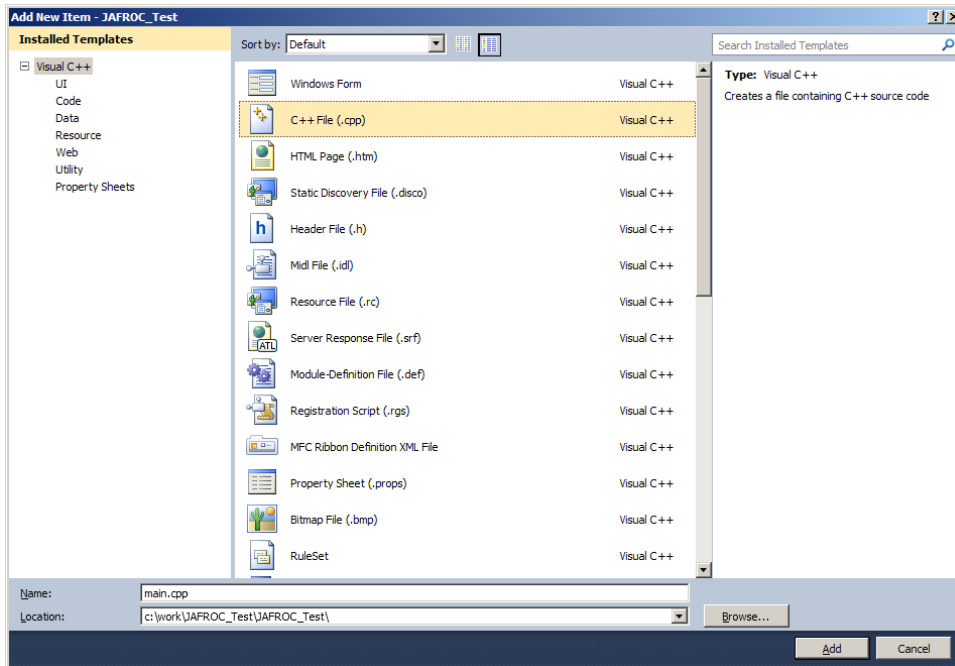


Start Coding

Copy a sample Excel formatted JAFROC data file to the project directory. For example, “DataFile.xls” included in JAFROC software. Also, “JAFROC_CORE.dll” file must be within the project folder. Copy it to a project folder.



In Visual Studio, select “Project → Add New Item” menu. Choose “C++ File (.cpp)”, assign a file name “main.cpp”, and press [Add] button.



Enter the following code to “main.cpp”.

```
#include <stdio.h>
#include "JAFROC_CORE.h"

int main( int argc, char *argv[] )
{
    int N_TREATMENTS, MAX_READERS, NCASES_NOR, NCASES_ABN, MAX_NL, MAX_LL;
    float ***NL, ***LL;
    int *N_Lesions_Per_Case;
    float **Weights;
    char **modalityTable, **readerTable;

    int i, j, k;
    int err;
    char errorString[ 200 ];

    float fom;

    float **TR_MEANS, ***PseudoValues;
    float MS_T, MS_R, MS_C, MS_TC, MS_RC, MS_TR, MS_TRC;
    float var_tr, var_tc, var_trc;

    float Results[ 2 ], DF[ 2 ];
    float Left[ 1 ], Right[ 1 ];
    float **TRT_CI;

    // read FROC data from Excel formatted file
    XLReader( "DataFile.xls", &N_TREATMENTS, &MAX_READERS,
              &NCASES_NOR, &NCASES_ABN, &MAX_NL, &MAX_LL,
              &NL, &LL, &N_Lesions_Per_Case, &Weights,
              &modalityTable, &readerTable, &err, errorString );

    // prepare memory space for analysis
    Allocate_analysis_memory( N_TREATMENTS, MAX_READERS, NCASES_NOR, NCASES_ABN,
                              &TR_MEANS, &PseudoValues, &TRT_CI );
}
```

```

// jackknifing routine is performed here
jackknife_FROC_Data( N_TREATMENTS, MAX_READERS, NCASES_NOR, NCASES_ABN,
                    MAX_NL, MAX_LL, NL, LL, N_Lesions_Per_Case, ANALYSIS_METHOD_JAFROC,
                    TR_MEANS, PseudoValues, &err );

// calculate mean-squares and variance components
mean_squares( N_TREATMENTS, MAX_READERS, NCASES_NOR, NCASES_ABN, PseudoValues,
              &MS_T, &MS_R, &MS_C, &MS_TC, &MS_RC, &MS_TR, &MS_TRC,
              &var_tr, &var_tc, &var_trc, &err );

// F-statistic and p-value are in Results[] array.
anova( N_TREATMENTS, MAX_READERS, NCASES_NOR, NCASES_ABN,
        TR_MEANS, PseudoValues, ANOVA_RANDOM_ALL,
        MS_T, MS_R, MS_C, MS_TC, MS_RC, MS_TR, MS_TRC,
        Results, DF, Left, Right, TRT_CI, &err );

// deallocate memory for FROC data
Free_analysis_memory( N_TREATMENTS, MAX_READERS,
                     TR_MEANS, PseudoValues, TRT_CI );

Free_FROC_Data( N_TREATMENTS, MAX_READERS, NCASES_NOR, NCASES_ABN, MAX_NL, MAX_LL,
                NL, LL, N_Lesions_Per_Case, Weights, modalityTable, readerTable );

return 0;
}

```

Select “Build → Build Solution” menu or press [F7] key to compile the sample code.

C API for JAFROC Development Kit

This section describes the API for calling the JAFROC_CORE.DLL. The following is the list of function calls and their arguments exported by the DLL. The DLL was compiled with Microsoft Visual C++ 2010. You may need to install “Microsoft Visual C++ 2010 Redistributable Package (x86)” if you don’t have “mfc100.dll” on your computer.

- **JAFROC_FOM function**

Returns JAFROC Figures-of-merit for a given FROC data set.

Syntax:

```

float JAFROC_FOM(
    int NCASES_NOR, int NCASES_ABN, int MAX_NL, int MAX_LL,
    float **NL, float **LL, int *N_Lesions_Per_Case,
    int analysis_method, int *err ) ;

```

Parameters:

- NCASES_NOR [in]
 - o int
 - o Number of normal cases
- NCASES_ABN [in]
 - o int
 - o Number of abnormal cases
- MAX_NL [in]
 - o int
 - o Maximum number of non-lesion localizations per case

- MAX_LL [in]
 - o int
 - o Maximum number of lesion localizations per case
- NL [in]
 - o float [NCASES_NOR + NCASES_ABN][MAX_NL]
 - o Non-lesion localizations contained in two-dimensional floating point valued array.
 - o Non-lesion localizations from normal cases must be located in lower indices.
 - o Empty array elements must be assigned UNINITIALIZED.
- LL [in]
 - o float [NCASES_ABN][MAX_LL]
 - o Lesion localizations contained in two-dimensional floating point valued array.
 - o Empty array elements must be assigned UNINITIALIZED.
- N_Lesions_Per_Case [in]
 - o int [NCASES_ABN]
 - o Number of lesions per each abnormal case.
- Analysis_method [in]
 - o int
 - o The method of calculating figure-of-merit and can be one of the following values

Value	Meaning
ANALYSIS_METHOD_JAFROC1	JAFROC1 figure-of-merit
ANALYSIS_METHOD_JAFROC	JAFROC figure-of-merit
ANALYSIS_METHOD_ROC	ROC figure-of-merit

- err [out]
 - o int *
 - o Error flag. If no error occurred, it returned 0. Otherwise, error is designated by its bit flags of one or more of the following

Error Flag	Meaning
JAFROC_DLL_ERROR_ZERO_N_ABN	Number of abnormal cases is set to zero.
JAFROC_DLL_ERROR_ZERO_N_NOR	Number of normal cases is set to zero. It won't be invoked in JAFROC1 analysis method.
JAFROC_DLL_ERROR_ZERO_N_MAX_NL	Number of non-lesion localization is set to zero.
JAFROC_DLL_ERROR_ZERO_N_MAX_LL	Number of lesion localization is set to zero.
JAFROC_DLL_ERROR_INVALID_METHOD	Analysis method is not valid.
JAFROC_DLL_ERROR_FLOAT_ERROR	INF and/or NaN occurred during floating-point arithmetic.

Return value:

Figure-of-merit of the given FROC data set with the assigned analysis method.

- **jackknife_FROC_Data function**

Perform a jackknifing process of the given FROC data.

Syntax:

```
void jackknife_FROC_Data(
    int N_TREATMENTS, int MAX_READERS, int NCASES_NOR, int NCASES_ABN,
    int MAX_NL, int MAX_LL, float ****NL, float ****LL,
    int N_Lesions_Per_Case, int analysis_method, float **TR_MEANS,
    float ***PseudoValues, int *err ) ;
```

Parameters:

- N_TREATMENTS [in]
 - o int
 - o Number of treatments.
- MAX_READERS [in]
 - o int
 - o Number of readers.
- NCASES_NOR [in]
 - o int
 - o Number of normal cases.
- NCASES_ABN [in]
 - o int
 - o Number of abnormal cases.
- MAX_NL [in]
 - o int
 - o Maximum number of non-lesion localizations per case.
- MAX_LL [in]
 - o int
 - o Maximum number of lesion localizations per case.
- NL [in]
 - o float [N_TREATMENTS][MAX_READERS][NCASES_NOR+NCASES_ABN][MAX_NL]
 - o Non-lesion localizations contained in four-dimensional floating point valued array.
 - o Non-lesion localizations from normal cases must be located in lower indices.
 - o Empty array elements must be assigned UNINITIALIZED.
- LL [in]
 - o float [N_TREATMENTS][MAX_READERS][NCASES_ABN][MAX_LL]
 - o Lesion localizations contained in two-dimensional floating point valued array.
 - o Empty array elements must be assigned UNINITIALIZED.
- N_Lesions_Per_Case [in]
 - o int [NCASES_ABN]
 - o Number of lesions per each abnormal case.
- Analysis_method [in]
 - o int
 - o The method of calculating figure-of-merit and can be one of the following values

Value	Meaning
ANALYSIS_METHOD_JAFROC1	JAFROC1 figure-of-merit
ANALYSIS_METHOD_JAFROC	JAFROC figure-of-merit
ANALYSIS_METHOD_ROC	ROC figure-of-merit

- TR_MEANS [out]
 - o float [N_TREATMENTS][MAX_READERS]
 - o Returned treat-means of the given FROC data set based on the assigned analysis method.
- PseudoValues [out]
 - o float [N_TREATMENTS][MAX_READERS][NCASES_NOR + NCASES_ABN]
 - o Returned pseudo-values of the given FROC data set based on the assigned analysis method.
- err [out]
 - o int *
 - o Error flag. If no error occurred, it returned 0. Otherwise, error is designated by its bit flags of one or more of the following

Error Flag	Meaning
JAFROC_DLL_ERROR_ZERO_N_ABN	Number of abnormal cases is set to zero.
JAFROC_DLL_ERROR_ZERO_N_NOR	Number of normal cases is set to zero. It won't be invoked in JAFROC1 analysis method.
JAFROC_DLL_ERROR_ZERO_N_MAX_NL	Number of non-lesion localization is set to

	zero.
JAFROC_DLL_ERROR_ZERO_N_MAX_LL	Number of lesion localization is set to zero.
JAFROC_DLL_ERROR_ZERO_READER	Number of readers is set to zero
JAFROC_DLL_ERROR_ZERO_TREATMENT	Number of treatments is set to zero
JAFROC_DLL_ERROR_INVALID_METHOD	Analysis method is not valid.
JAFROC_DLL_ERROR_FLOAT_ERROR	INF and/or NaN occurred during floating-point arithmetic.

- **Mean Squares function**

Calculates mean squares of the FROC data based on its pseudo-values determined by the jackknife_FROC_Data() function.

Syntax:

```
void mean_squares(
    int N_TREATMENTS, int MAX_READERS, int NCASES_NOR, int NCASES_ABN,
    float ***PseudoValues, float *OUT_MS_T, float *OUT_MS_R,
    float *OUT_MS_C, float *OUT_MS_TC, float *OUT_MS_RC,
    float *OUT_MS_TR, float *OUT_MS_TRC, float *OUT_var_tr,
    float *OUT_var_tc, float *OUT_var_trc, int *err );
```

Parameters:

- N_TREATMENTS [in]
 - o int
 - o Number of treatments.
- MAX_READERS [in]
 - o int
 - o Number of readers.
- NCASES_NOR [in]
 - o int
 - o Number of normal cases.
- NCASES_ABN [in]
 - o int
 - o Number of abnormal cases.
- PseudoValues [in]
 - o float [N_TREATMENTS][MAX_READERS][NCASES_NOR + NCASES_ABN]
 - o Pseudo-values of the FROC data set obtained by the jackknife_FROC_Data() function.
- OUT_MS_T [out]
 - o float *
 - o Returned mean square for treatments.
- OUT_MS_R [out]
 - o float *
 - o Returned mean square for readers.
- OUT_MS_C [out]
 - o float *
 - o Returned mean square for cases.
- OUT_MS_TC [out]
 - o float *
 - o Returned mean square for treatments and cases.
- OUT_MS_RC [out]
 - o float *
 - o Returned mean square for treatments and readers.
- OUT_MS_TR [out]
 - o float *
 - o Returned mean square for treatments and readers.

- OUT_MS_TRC [out]
 - o float *
 - o Returned mean square for treatments, readers, and cases.
- OUT_var_tr [out]
 - o float *
 - o Returned variance components for treatments and readers.
- OUT_var_tc [out]
 - o float *
 - o Returned variance components for treatments and cases.
- OUT_var_trc [out]
 - o float *
 - o Returned variance components for treatments, readers, and cases.
- err [out]
 - o int *
 - o Error flag. If no error occurred, it returned 0. Otherwise, error is designated by its bit flags of one or more of the following

Error Flag	Meaning
JAFROC_DLL_ERROR_ZERO_N_ABN	Number of abnormal cases is set to zero.
JAFROC_DLL_ERROR_ZERO_READER	Number of readers is set to zero
JAFROC_DLL_ERROR_ZERO_TREATMENT	Number of treatments is set to zero
JAFROC_DLL_ERROR_FLOAT_ERROR	INF and/or NaN occurred during floating-point arithmetic.

- **ANOVA function**

Syntax:

```
void anova(
    int N_TREATMENTS, int MAX_READERS, int NCASES_NOR, int NCASES_ABN,
    float **TR_MEANS, float ***PseudoValues, int anova_random,
    float MS_T, float MS_R, float MS_C, float MS_TC, float MS_RC,
    float MS_TR, float MS_TRC, float *Results, float *DF, float *Left,
    float *Right, float **TRT_CI, int *err ) ;
```

Parameters:

- N_TREATMENTS [in]
 - o int
 - o Number of treatments.
- MAX_READERS [in]
 - o int
 - o Number of readers.
- NCASES_NOR [in]
 - o int
 - o Number of normal cases.
- NCASES_ABN [in]
 - o int
 - o Number of abnormal cases.
- TR_MEANS [in]
 - o float [N_TREATMENTS][MAX_READERS]
 - o Treatment means of the FROC data set obtained by the jackknife_FROC_Data() function.
- PseudoValues [in]
 - o float [N_TREATMENTS][MAX_READERS][NCASES_NOR + NCASES_ABN]
 - o Pseudo-values of the FROC data set obtained by the jackknife_FROC_Data() function.
- anova_random [in]
 - o int
 - o Determines which method ANOVA runs. Can be one of the following values

Value	Meaning
ANOVA_RANDOM_ALL	Performing ANOVA with respect to all
ANOVA_RANDOM_CASES	Performing ANOVA with respect to cases
ANOVA_RANDOM_READERS	Performing ANOVA with respect to readers

- MS_T [in]
 - o float *
 - o Mean square for treatments.
- MS_R [in]
 - o float *
 - o Mean square for readers.
- MS_C [in]
 - o float *
 - o Mean square for cases.
- MS_TC [in]
 - o float *
 - o Mean square for treatments and cases.
- MS_RC [in]
 - o float *
 - o Mean square for treatments and readers.
- MS_TR [in]
 - o float *
 - o Mean square for treatments and readers.
- MS_TRC [in]
 - o float *
 - o Mean square for treatments, readers, and cases.
- Results [out]
 - o float [2]
 - o Returned F-statistics in the first element ([0]) and p-values in the second element ([1]).
- DF [out]
 - o float [2]
 - o Returned degree-of-freedom, numerator in [0] and denominator in [1].
- Left [out]
 - o float [N_TREATMENTS * (N_TREATMENTS - 1) / 2]
 - o Lower bounds of 95% confidence intervals of inter-treatment differences
- Right [out]
 - o float [N_TREATMENTS * (N_TREATMENTS - 1) / 2]
 - o Upper bounds of 95% confidence intervals of inter-treatment differences
- TRT_CI [out]
 - o float [N_TREATMENTS][2]
 - o Lower and upper bounds of 95% confidence intervals of the reader-averaged FOMs.
 - o Lower bounds for each treatment are in [0] and upper bounds are in [1]
- err [out]
 - o int *
 - o Error flag. If no error occurred, it returned 0. Otherwise, error is designated by its bit flags of one or more of the following

Error Flag	Meaning
JAFROC_DLL_ERROR_ZERO_N_ABN	Number of abnormal cases is set to zero.
JAFROC_DLL_ERROR_ZERO_READER	Number of readers is set to zero
JAFROC_DLL_ERROR_ZERO_TREATMENT	Number of treatments is set to zero
JAFROC_DLL_ERROR_FLOAT_ERROR	INF and/or NaN occurred during floating-point arithmetic.
JAFROC_DLL_ERROR_INVALID_ANOVA	Performing ANOVA with improper data set. (e.g. ANOVA random readers of single-reader data set)

- **XLReader function**

XLReader() function reads Microsoft Excel formatted JAFROC data file and returns data array along with its number of elements and labels of reader and modality ID's by string.

Syntax:

```
void __cdecl XLReader(
    char *filename, int *N_TREATMENTS, int *MAX_READERS,
    int *NCASES_NOR, int *NCASES_ABN, int *MAX_NL, int *MAX_LL,
    float *****NL, float *****LL, int **N_Lesions_Per_Case,
    float ***Weights, char ***readerTable, char ***modalityTable,
    int *err, char *errorString ) ;
```

Parameters:

- filename [in]
 - o char *
 - o Excel formatted JAFROC data file name to be read.
- N_TREATMENTS [out]
 - o int *
 - o Number of modalities in the Excel formatted JAFROC data file.
- MAX_READERS [out]
 - o int *
 - o Number of readers in the Excel formatted JAFROC data file.
- NCASES_NOR [out]
 - o int *
 - o Number of normal cases in the Excel formatted JAFROC data file.
- NCASES_ABN [out]
 - o int *
 - o Number of abnormal cases in the Excel formatted JAFROC data file.
- MAX_NL [out]
 - o int *
 - o Maximum number of non-lesion localizations in the Excel formatted JAFROC data file.
- MAX_LL [out]
 - o int *
 - o Maximum number of lesion localizations in the Excel formatted JAFROC data file.
- NL [out]
 - o &float [N_TREATMENTS][MAX_READERS][NCASES_NOR+NCASES_ABN][MAX_NL]
 - o Non-lesion localization ratings in the Excel formatted JAFROC data file.
- LL [out]
 - o &float [N_TREATMENTS][MAX_READERS][NCASES_ABN][MAX_LL]
 - o Lesion localization ratings in the Excel formatted JAFROC data file.
- N_Lesions_Per_Case [out]
 - o &int [NCASES_ABN]
 - o Number of lesion localizations per each abnormal case in the Excel formatted JAFROC data file.
- Weights [out]
 - o &float [NCASES_ABN][MAX_LL]
 - o Weights of the lesion localization marks defined in the Excel formatted JAFROC data file.
- readerTable [out]
 - o &char *[MAX_READERS]
 - o Reader IDs assigned in the Excel formatted JAFROC data file.
- modalityTable [out]
 - o &char *[N_TREATMENTS]

- o Modality IDs assigned in the Excel formatted JAFROC data file.
- err [out]
 - o int *
 - o Error flag. If no error occurred, it returned 0. Otherwise, error is designated by its bit flags.

Error Flag	Meaning
JAFROC_DLL_ERROR_XL_FILE_READ	Error reading JAFROC data file.

- errorString [out]
 - o char [200]
 - o Returned error description in detail.

- **Free_FROC_Data function**

Syntax:

```
JAFROC_CORE_API void __cdecl Free_FROC_Data(
    int N_TREATMENTS, int MAX_READERS, int NCASES_NOR, int NCASES_ABN,
    int MAX_NL, int MAX_LL, float ****NL, float ****LL,
    int *N_Lesions_Per_Case, float **Weights,
    char **readerTable, char **modalityTable ) ;
```

Parameters:

- N_TREATMENTS [in]
 - o int
 - o Number of modalities.
- MAX_READERS [in]
 - o int
 - o Number of readers.
- NCASES_NOR [in]
 - o int
 - o Number of normal cases.
- NCASES_ABN [in]
 - o int
 - o Number of abnormal cases.
- MAX_NL [in]
 - o int
 - o Maximum number of non-lesion localizations.
- MAX_LL [in]
 - o int
 - o Maximum number of lesion localizations.
- NL [in]
 - o float
 - o [N_TREATMENTS][MAX_READERS][NCASES_NOR+NCASES_ABN][MAX_NL]
 - o Non-lesion localization ratings array to be de-allocated.
- LL [in]
 - o float [N_TREATMENTS][MAX_READERS][NCASES_ABN][MAX_LL]
 - o Lesion localization ratings array to be de-allocated.
- N_Lesions_Per_Case [in]
 - o int [NCASES_ABN]
 - o Number of lesion localizations per each abnormal case array to be de-allocated.
- Weights [in]
 - o float [NCASES_ABN][MAX_LL]
 - o Weights of the lesion localization marks to be de-allocated.
- readerTable [in]
 - o char *[MAX_READERS]

- Reader IDs string array to be de-allocated.
- modalityTable [in]
 - char *[N_TREATMENTS]
 - Modality IDs string array to be de-allocated.

- **Allocate_analysis_memory function**

Syntax:

```
void Allocate_analysis_memory(
    int N_TREATMENTS, int MAX_READERS, int NCASES_NOR, int NCASES_ABN,
    float ***TR_MEANS, float ****PseudoValues, float ***TRT_CI ) ;
```

Parameters:

- N_TREATMENTS [in]
 - int
 - Number of modalities.
- MAX_READERS [in]
 - int
 - Number of readers.
- NCASES_NOR [in]
 - int
 - Number of normal cases.
- NCASES_ABN [int]
 - int
 - Number of abnormal cases.
- TR_MEANS [out]
 - &float **
 - Memory allocated to handle treatment means of the data set. Size of the array is [N_TREATMENTS][MAX_READERS].
- PseudoValues [out]
 - &float ***
 - Memory allocated to handle pseudo-values of the data set. Size of the array is [N_TREATMENTS][MAX_READERS][NCASES_NOR+NCASES_ABN].
- TRT_CI [out]
 - &float **
 - Memory allocated to handle confidence intervals of the reader-averaged FOMs. Size of the array is [N_TREATMENTS][2].

- **Free_analysis_memory function**

Syntax:

```
void Free_analysis_memory(
    int N_TREATMENTS, int MAX_READERS, float **TR_MEANS,
    float ****PseudoValues, float **TRT_CI ) ;
```

Parameters:

- N_TREATMENTS [in]
 - int
 - Number of modalities.
- MAX_READERS [in]
 - int
 - Number of readers.
- TR_MEANS [in]
 - float **

- Treatment means array to be de-allocated.
- PseudoValues [in]
 - float ***
 - Pseudo-values array to be de-allocated.
- TRT_CI [in]
 - float **
 - Confidence intervals array to be de-allocated.

Last modification date: 1/18/2011