|  |  |
| --- | --- |
|  | Overview  Version 2.0.Beta1 - April 2011 |

Abstract

The Microsoft Biology Foundation (MBF) is an open source, reusable .NET library and application programming interface (API) for bioinformatics research.

This document gives an overview of MBF, its components and tools.

The Microsoft Biology Foundation is available at http://mbf.codeplex.com.

Contents

[Introduction 3](#_Toc289675293)

[What is MBF 3](#_Toc289675294)

[An open source project for community participation 3](#_Toc289675295)

[Contribution Roles 4](#_Toc289675296)

[Benefits of MBF 5](#_Toc289675297)

[Reusable libraries 5](#_Toc289675298)

[Use any application style 5](#_Toc289675299)

[ Cross platform deployment 5](#_Toc289675300)

[Perform a wide range of tasks 6](#_Toc289675301)

[Implementation in any .NET compatible language 6](#_Toc289675302)

[What’s New and Changed 6](#_Toc289675303)

[Change list 6](#_Toc289675304)

[How to Install MBF 8](#_Toc289675305)

[Prerequisites 8](#_Toc289675306)

[System Requirements 8](#_Toc289675307)

[Installation 9](#_Toc289675308)

[Migration to Newer Versions 10](#_Toc289675309)

[Installer Behavior 10](#_Toc289675310)

[DLL versioning 11](#_Toc289675311)

[The MBF Architecture 11](#_Toc289675312)

[MBF Components 13](#_Toc289675313)

[The Project Samples 15](#_Toc289675314)

[Resources 16](#_Toc289675315)

Disclaimer: This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Visual Studio, and Windows are trademarks of the Microsoft group of companies.  All other trademarks are property of their respective owners.

# Introduction

This document gives an overview of the Microsoft Biology Foundation (MBF), an open source, reusable .NET Framework library and application programming interface (API) for bioinformatics research. It is designed to encourage extension, reuse, and community contribution via release as part of the CodePlex Open Source Initiative (OSI) at Microsoft.

## What is MBF

MBF is essentially a bioinformatics toolkit built on top of the .NET Framework 4.0 providing a foundation upon which other tools can be built. It is designed to manipulate large data sets by using in-memory compression of sequence data and scalable algorithms that take advantage of multiple cores and provide numerous components for biological analysis including:

* Parsers/formatters to read/write common bioinformatics formats
* support for DNA, RNA and protein sequences
* algorithm framework for analysis and transformation
* web connector framework for web-service interaction

MBF is primarily focused on genomics with the following:

* reusable data structures to represent sequences and symbols
* I/O framework to load and save sequences
* algorithm framework to process loaded sequences

## An open source project for community participation

The primary goals have been to enable participation by the bioinformatics community and to obtain a better technical understanding of the underlying object model, extensibility, and code architecture requirements to meet the needs of this community. To this end, MBF is available under an open source license with the two levels of participation described in the [contribution roles](#_Contribution_Roles) section. Executables, source code, demo applications, and documentation are freely downloadable from the following web sites:

* <http://research.microsoft.com/bio> for executables and training materials
* [http//mbf.codeplex.com](file:///C:\Users\michaelz\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\Content.Outlook\WYRO2NCW\http\mbf.codeplex.com) for source code and documentation

We encourage you to provide feedback on MBF at <http://mbf.codeplex.com>.

With this in mind, the project has been designed with the following characteristics:

Design Characteristics

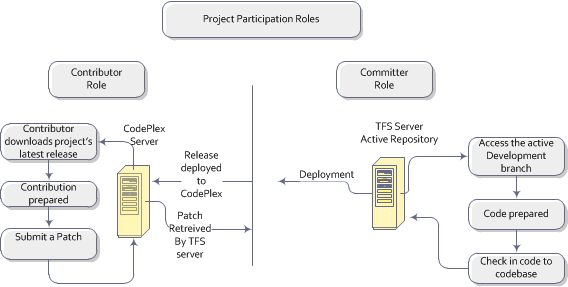
|  |  |
| --- | --- |
| Design | Description |
| Extensibility | Extensibility is an integral part of the design. Core concepts are mapped as interfaces and ABCs enabling you to easily provide alternative implementations or add any additional features you need. |
| Language neutrality | The project is built on top of .NET enabling you to use any .NET supported language including support for dynamic languages such as IronPython. |
| Best practices | Best practices are observed throughout the project. Source code is commented and algorithms all cite publications so underlying principles and assumptions are accounted for and nothing is a black box . |
| Interoperability | The code can be run on several mainstream platforms |

# Contribution Roles

There are two avenues of participation in the project:

* As a Contributor – accessing the code base deployed on Codeplex as self-contained .zip file and submitting your work through CodePlex. You must first register an account on CodePlex.
* As a Committer – directly accessing the code base in the active development repository and directly checking in your changes to the repository. You must have committer rights to the project and Microsoft extranet partner account that belongs to the VSTS\_MBI\_Extranet\_Contributors security group.

The following figure illustrates the two participation roles.



While contributors only have access to the periodically released code and can only submit candidate code through the Codeplex **Submit a patch** feature, committers have direct access to the active code base for the project for both code check-outs and check-ins.

Details for each role are provided in the Contributor Guide and the Committers Guide <http://mbf.codeplex.com/documentation>.

# Benefits of MBF

The bioinformatics field is still relatively young. The term ‘bioinformatics’ dates back to just the mid-1990s, and its practitioners come from a wide variety of backgrounds, particularly biology, physics, chemistry, and occasionally computer science.

Industry-standard architectural and coding practices are not consistently used by the bioinformatics community. Data formats are still poorly defined, with a number of emerging schemas capturing only some of the details. Scalability is becoming a serious problem as bioinformatics datasets grow in size. Simplistic implementations fail to execute on single CPUs and also have no provisions for leveraging multicore or multiprocessor architectures.

## Reusable libraries

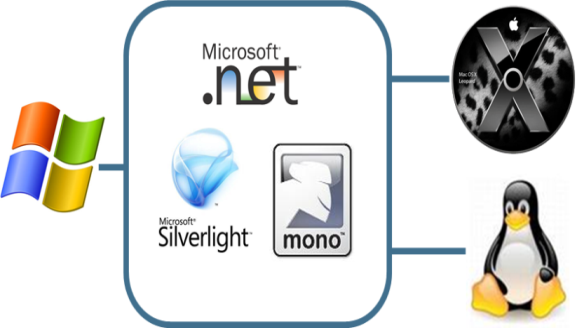
For these reasons, the need has been recognized for reusable libraries of high-quality bioinformatics code. The goal for MBF is to provide a development framework for the bioinformatics community that ensures the high architectural and coding standards necessary for ease of application extensibility and longevity.

## Use any application style

You can develop console applications, NT Services, Graphical User Interfaces (GUI) using WIN Forms, dynamic and interactive ASP.NET web pages, in the cloud applications via the Azure cloud computing platform, and Web based service applications using Silverlight as depicted in the following figure:

## Cross platform deployment

To use MBF in other environments, use Silverlight as your application style and then rely on Mono/Moonlight or the Silverlight plug-in on the OSX platform to run your code. You can use Visual Studio for your IDE on the Windows platform, or if you are working with the source code, you can use a Mono-based IDE such as MonoDevelop or SharpDevelop.



**Note**: Mono is open source .Net that runs on Linux. Silverlight is browser plug-in using a subset of .Net that supports most popular browsers including Internet Explorer, Chrome, Firefox and Safari. Moonlight is built on top of Mono.

## Perform a wide range of tasks

You can use MBF to perform a wide range of tasks, including:

* Assembly of Large Genome: This is where the researcher would like to use the MBI (library and tools) to assemble a large Genome Browser
* Import DNA, RNA, or protein sequences from files with a variety of standard data formats, including FASTA, FASTQ, GFF, GenBank, and BED.
* Construct sequences from scratch.
* Manipulate sequences in various ways, such as adding or removing elements or generating a complement.
* Analyze sequences using algorithms such as Smith-Waterman and Needleman-Wunsch.
* Submit sequence data to remote Web sites—such as a Basic Local Alignment Search Tool (BLAST) Web site—for analysis.
* Output sequence data in any supported file format, regardless of the input format.

## Implementation in any .NET compatible language

MBF applications can be implemented in any of over 70 .NET compatible languages, including C#, F#, Visual Basic® .NET, and IronPython. Programming guides at <http://mbf.codeplex.com/documentation>, describe how to develop MBF applications using C# and IronPython.

# 

# What’s New and Changed

The current version, v2.0.Beta1, fixes bugs, implements new features and improves existing features.

## Change list

The following table contains the change list summary for this release. For more details and a list of API changes go to the Programming Guide v2.

Change list for 2.0.Beta1

|  |  |
| --- | --- |
| Change | Description |
| [Comparative Assembly](#_Comparative_Assembly) | New assembly techniques provided to allow for re-sequencing and comparative assembly of genomes to a reference of same species or of similar species. |
| ComparativeUtil | **New** - A utility to kick off the comparative assembly. |
| ConsensusUtil | **New** - Used for ComparativeUtil step 4. Users can manipulate the data before using it as an input for the next step in the chain. |
| Fasta - > FastA | To be consistent with the industry norm. |
| Installation | All sample tools (i.e., BioExcel, Sequence Assembler, etc..) now also licensed under Apache 2.0 and distributed as one installation with the Bio library |
| LayoutRefinementUtil | **New** - Used for ComparativeUtil step 3. Users can manipulate the data before using it as an input for the next step in the chain. |
| LISUtil | **New** - A utility tool for the longest increasing sequence of mummer |
| MUMmer | Optimizations to support large genome assembly. |
| New License | Entire library moved from MS-PL to more commonly accepted Apache 2.0 OSI approved license. |
| New namespace | MBF namespace has been renamed Bio. |
| NucmerUtil | **New** - Used for ComparativeUtil step 1. Users can manipulate the data before using it as an input for the next step in the chain. |
| Optimization work | a) Memory profiling and analysis on the framework optimized.  b) Padena memory optimizations.  c) Sequence optimizations, including non-string and non-character sequences.  d) MUMmer optimization based on suffix tree and links improved.  e) Object Model optimizations.  f) More scenarios for collection of memory and performance profiles. |
| Padena assembly algorithm | Increased capacity to support assembly of large genomes. Performance improvements to De Bruijn graph generation. |
| PadenaUtil | A utility that defines the scaffolding. |
| Parser and formatter Encoding | **Removed** - Parsers and formatters no longer take encodings. We removed the whole encoding class. |
| RepeatResolutionUtil | **New** - Used for ComparativeUtil step 2. Users can manipulate the data before using it as an input for the next step in the chain. |
| SAMUtils | A sequence coverage analysis utility. |
| ScaffoldUtil | **New** - Used for ComparativeUtil step 5. Users can manipulate the data before using it as an input for the next step in the chain. |
| Sequence Object Model | a) Re-designed to be much more memory efficient.  b) Use of condensed binary representation of DNA, RNA and Proteins rather than text characters.  c) Improved the capacity by using **IEnumerable<byte**>. **Dictionary** and **hashset** are used to store items such as ambiguous alphabets.  d) More efficient implementation of encodings.  e) encoding removed from sequence object model (parsers and formatters).  f) Changes to use **ISequence** : **IList<byte>.** |
| Source tree changes | MBF\Source\MBF -> Bio\Source\Framework  MBF\Source\MBF - > Bio\Source\Tools |
| Data Virtualization | **Removed** |

Note: Documentation can be found in the same folder as the utility source code at ..\MBI\Source\MBT.

# How to Install MBF

This section describes the prerequisites, system requirements, and installation steps for the project.

## Prerequisites

To use the basic capabilities of MBF, you should have a basic understanding of:

* Methods and nomenclature of genomics and bioinformatics.
* Productivity software similar to Microsoft Office Excel.

To take advantage of programming and extensibility capabilities, you need at least:

* Basic programming skills.
* Familiarity with using Microsoft Visual Studio® to program .NET applications with C#.
* Basic understanding of programming for Web services.

## System Requirements

* Windows® XP Service Pack (SP) 2 and later versions of Windows.
* .NET Framework Version 4.0, available at <http://go.microsoft.com/fwlink/?LinkID=186913>

Additional software requirements for implementing MBF applications are described in “Microsoft Biology Foundation Programming Guide.”

## Installation

The MBI project also periodically posts stable snapshots of the source tree to the [MBI CodePlex Web](http://mbf.codeplex.com/) site. If you are interested in MBF but do not want to submit code to the repository, you can obtain a copy of the source tree by downloading a snapshot. You do not need committer or contributor status for such downloads, the snapshots are available to any interested user.

More details and the installer, Setup\_Framework.msi, are available at <http://mbf.codeplex.com>.

To install MBF

1. Copy the MBF installer, Bio.msi, to a folder on your hard drive.
2. Navigate to that folder and double-click Bio.msi, which runs the installation wizard.
3. Follow the directions in the installation wizard to install the project.

**Note**: Select **Complete** install to install the SDK.

The MBF installer creates a directory named C:\Program Files (x86)\Microsoft Biology Initiative\2.0\MBF that contains the following:

\Add-ins

\Bio.Comparative.dll

\Bio.Padena.dll

\Bio.Pamsam.dll

Bio.dll

Bio.WebServiceHandlers.dll

Readme.txt

If you install the optional SDK, the installer creates an SDK folder in the Bio folder that contains the following:

\SDK

\Samples

\ComparativeUtil

\ConsensusUtil

\IronPython

\LayoutRefinementUtil

\LISUtil

\MumUtil

\NucmerUtil

\PadenaUtil

\ReadSimulator

\RepeatResolutionUtil

\SAMUtils

\ScaffoldUtil

\TridentWorkflows

Bio.chm

Coding\_Conventions.docx

Commenting\_Conventions.docx

Committer\_Guide

Contribution\_Documentation\_Template.docx

Contribution\_Guide.docx

Getting\_Started.docx

Becoming\_A\_Committer

Onboarding.docx

Overview.docx

PaDeNa.docx

Programming\_Guide.docx

IronPython\_Programming Guide.docx

Testing\_Guide.docx

**Note** : To get the API documentation to appear in the Intellisense pop ups you must build the project to create the XML documentation file then make sure that file resides in the same folder where the MBF.dll is installed.

# Migration to Newer Versions

For those that have a previous version of MBF installed you are not required to uninstall it in order to install a newer version. Versions can be installed side-by-side.

Previous versions:  
v0.2 (Beta),   
v1.0   
v2.1 (Dev Preview).

The current version  
v2.0   
**Note** the previous release, v2.1 (Dev Preview) was given an inappropriate version number. Therefore 2.0 will logically be treated to be a higher version than 2.1.

## Installer Behavior

The expected installer behavior in cases of a previously installed release is as follows:

* Silent upgrade on any minor version change.
* Side-by-side installation for any major version change (after showing a message saying a previous version exists and suggesting uninstalling it).

The following are the behaviors for specific preexisting installs:

* If v0.2 and/or v1.0  are installed, a silent side-by-side installation is performed.
* If v2.1(Dev Preview) is installed, a side-by-side installation with a warning message is performed.

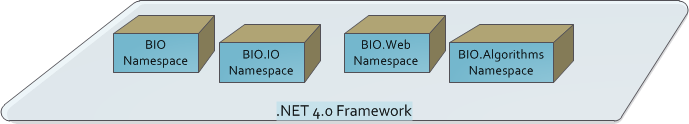
## DLL versioning

The following dll versioning conventions are observed:

* All releases of MBF DLL’s prior to the current version, v2.0, are versioned as 1.0.0.0
* Any DLL version inside a package has no relation with its product version. For example, MyProduct v3.0 can have DLL’s which are v1.0 / v5.0.
* MBF v2.0 has all DLL versions as 2.0.

# The MBF Architecture

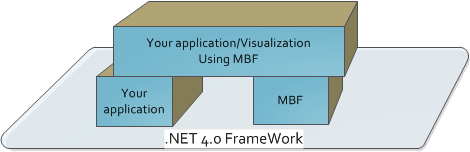
MBF is essentially a bioinformatics toolkit built on top of the .NET Framework 4.0 providing a foundation upon which other tools can be built. The project contains libraries of reusable bioinformatics functions and algorithms using the .NET Framework. Each library has its own namespace and associated classes. The following figure shows the namespace for each of the four libraries.



The namespaces provide the following components to support the biological analysis process. See BioDotNet.chm in the SDK folder for a complete reference.

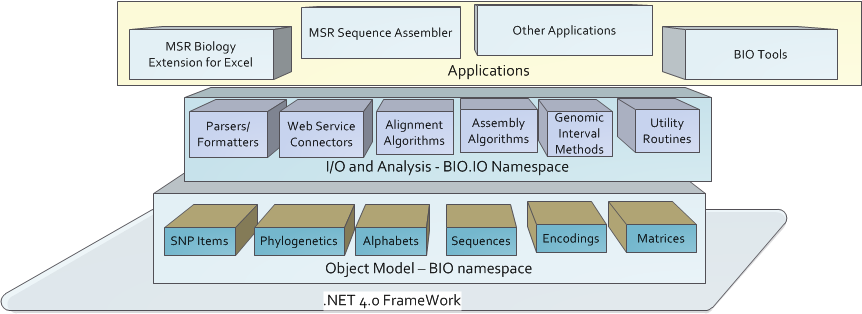
* **BIO:** An object model for storing sequence data, metadata, and encodings
* **BIO.Web**: Web services interface for connecting object model to various web-based components. BLAST and ClustalW are default implementations.
* **BIO.Algorithms**: Algorithms for translation, pair-wise and multi-sequence alignment, and sequence assembly.
* **BIO.IO**: Parsers and formatters for various forms of genomic data.

MBF is not an application in itself. It does not provide any visualization of the data rather it provides the basis for visualizations to be built on top of as illustrated in the following figure.



Using MBF is as simple as adding a reference to Bio.dll to your project. You can then begin using the available types.

Sequences are the core concept in MBF. They contain [symbols] based on alphabets, are read and written using parsers and formatters, passed as arguments and returned from algorithms. The following figure illustrates the overall project architecture.



**Note**: Assemblers and aligners are supported as add-ins. Decorate the class with attributes and add-in.

The following figure illustrates the basic work flow.



## MBF Components

MBF implements:

* An object model for representing genomic data.
* A range of parsers for common bioinformatics file formats.
* A range of algorithms for manipulating DNA, RNA, and protein sequences.
* A set of connectors to biological Web services such as NCBI BLAST.

You can also work with sequences using two tools included with deployed project: Microsoft Research Biology Extension for Excel, an add-in for Microsoft Excel, and MSR Sequence Assembler, a .NET application. For more information, see the accompanying documents in the ..\Bio\Doc document folder:

### Tools

The following tools, located at ..\Bio\Source\Tools, are included in the deployed project:

Tools 2.0.Beta1

|  |  |
| --- | --- |
| Tool | Description |
| Microsoft Research Sequence Assembler | Provides a packaged sequence assembler GUI. MSR\_Sequence\_Assembler\_User\_Guide.docx |
| Microsoft Research Biology Extension for Excel | Provides an Excel toolbar ribbon to access MBF functionality. MBF\_Biology\_Extension\_User\_Guide.docx |

### Parsers and Formatters

The following parsers and formatters are included in the deployed project types (at ..\Bio\Source\Framework\Bio\IO):

|  |  |  |
| --- | --- | --- |
| Formats | Parser or Formatter | Description |
| FastA | Parser and Formatter | Sequence format |
| FastQ | Parser and Formatter | Sequence format |
| GenBank | Parser and Formatter | Sequence format |
| GFF | Parser and Formatter | Sequence format |
| Newick | Parser and Formatter | Phylogenetics |
| Nexus | Parser | Sequence alignment |
| Phylip | Parser | Phylogenetics |
| SAM and BAM | Parser and Formatter | Sequence alignment |
| BED | Parser and Formatter | Sequence format |
| ClustalW | Parser | Sequence alignment |
| snpParser and SimplesnpParser | Parser |  |
| XSV related Parser and formatters |  |  |
|  |  | XsvTextReader  XsvSparseReader  XsvSparseParser  XsvSparseFormatter  XsvSnpReader  XsvContigParser  XsvContigFormatter |

### Web Services

The following web services and their service handlers are included in the deployed project:

|  |  |
| --- | --- |
| Web Services | Description |
| Azure | ..\Bio\Source\Framework\Bio.WebServiceHandlers |
| BioHPC | ..\Bio\Source\Framework\Bio.WebServiceHandlers |
| EDI | ..\Bio\Source\Framework\Bio.WebServiceHandlers |
| NCBI | ..\Bio\Source\Framework\Bio.WebServiceHandlers |
| BLAST | Handler MBF.Web.Blast.IBlastServiceHandler at ..\Bio\Source\Framework\Bio\Web. |
| ClustalW | Handler MBF.Web.ClustalW.IClustalWServiceHandler at ..\Bio\Source\Framework\Bio\Web. |

### Built-in Algorithm Aligners

Implementations of several standard algorithms are included in the deployed project including the following algorithm aligners (at ..\Bio\Source\Framework\Bio\Algorithms\Alignment):

|  |  |
| --- | --- |
| Algorithm aligners | Description |
| PairwiseOverlapAligner | Reference implementation of a simple pairwise overlap algorithm for aligning two sequences. |
| NeedlemanWunschAligner | Global alignment (where entire sequence is compared) using Needleman-Wunsch algorithm. |
| SmithWatermanAligner | Local alignment (where partial sequences are compared) using Smith-Waterman algorithm. |
| MUMmerAligner | Algorithm used for aligning entire genomes or very large protein sequences. It in turn calls MUMmer. |
| NucmerPairwiseAligner | Algorithm used for aligning entire genomes or very large DNA sequences. |

# The Project Samples

The project includes sample code and sample data files to help get you started. The project includes the following samples:

|  |  |
| --- | --- |
| Sample application | Description |
| AlignSequences | Located in the Programming Guide.docx. It demonstrates perform sequence alignment and use how to use **SequenceStatistics** to iterate through the sequence. |
| Bio.Workflow | By default deployed to C:\Program Files (x86)\Microsoft Biology Initiative\2.0\Tools and the source code is located at $/MBI/DevelopmentBranch/BIO/SourceSamples. |
| BioDemo.py | An IronPython demonstration of some of the current non-GUI features.  By default deployed to C:\Program Files (x86)\Microsoft Biology Initiative\2.0\Tools and the source code is located at $/MBI/DevelopmentBranch/BIO/SourceSamples. |
| BlastRequest | Located in the Programming Guide.docx. It demonstrates how to use Blast and how to use **WebRequest**. |
| GenBank Data file | A sample data file included with the Programming Guide. It demonstrates how to work with a sequence segment including how to reverse and complement the sequence. |
| ManipulateSequence | Located in the Programming Guide.docx. |
| ReadSimulator | By default deployed to C:\Program Files (x86)\Microsoft Biology Initiative\2.0\Tools and the source code is located at $/MBI/DevelopmentBranch/BIO/SourceSamples. |

**Note**: Documentation for each sample is located in the same folder as the sample.

There is also a Starter Project and several sample data files in the training material at <http://research.microsoft.com/bio> which guides you through a new C# project.

# Resources

This section provides links to additional information about MBF and related topics.

Microsoft Resources

IronPython

<http://www.codeplex.com/IronPython/>

Microsoft Biology Foundation at Microsoft Research

<http://research.microsoft.com/en-us/collaboration/tools/mbf.aspx>

Visual Studio 2010 and .NET Framework 4

<http://msdn.microsoft.com/vstudio/>

CodePlex Resources

Microsoft Biology Framework

* [http://mbf.codeplex.com/](http://research.microsoft.com/en-us/projects/bio/mbf.aspx)  
  Microsoft Biology Foundation: Overview   
  Microsoft Biology Foundation: Programming Guide   
  Microsoft Research Sequence Assembler: User Guide  
  Padena: Parallel DeNovo Assembler

Research Biology Extension for Excel User’s Guide

<http://bioexcel.codeplex.com/>  
Microsoft Biology Extension for Excel: User Guide

Sandcastle

Sandcastle - Documentation Compiler for Managed Class Libraries  
<http://sandcastle.codeplex.com/>   
Sandcastle Help File Builder  
<http://www.codeplex.com/SHFB>

Bioinformatics References

BLAST

<http://blast.ncbi.nlm.nih.gov/Blast.cgi>

EBI BLAST Service

<http://www.ebi.ac.uk/Tools/blast2/index.html>

FASTA format description

<http://www.ncbi.nlm.nih.gov/blast/fasta.shtml>

FASTQ format description

<http://maq.sourceforge.net/fastq.shtml>

GenBank

Overview   
<http://www.ncbi.nlm.nih.gov/Genbank/>  
Sample GenBank Record   
<http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html>

GFF Specification

<http://www.sanger.ac.uk/resources/software/gff/spec.html>

International Nucleotide Sequence Database Collaboration

http://insdc.org/favicon.ico

National Center for Biotechnology Information

[http://www.ncbi.nlm.nih.gov](http://www.ncbi.nlm.nih.gov/)