|  |  |
| --- | --- |
|  | .NET Bio Framework Overview  Version 1.0 - July 2011 |

Abstract

The .NET Bio Framework is an open source, reusable .NET library and application programming interface (API) for bioinformatics research.

This document gives an overview of .NET Bio Framework, its components and tools.

The Framework is available at <http://bio.codeplex.com>.

Contents

[Introduction 3](#_Toc294083057)

[What is .NET Bio Framework 3](#_Toc294083058)

[An open source project for community participation 3](#_Toc294083059)

[Contribution Roles 4](#_Toc294083060)

[Benefits of .NET Bio Framework 5](#_Toc294083061)

[Reusable libraries 5](#_Toc294083062)

[Use any application style 5](#_Toc294083063)

[Cross platform deployment 5](#_Toc294083064)

[Perform a wide range of tasks 6](#_Toc294083065)

[Implementation in any .NET compatible language 6](#_Toc294083066)

[What’s New and Changed 6](#_Toc294083067)

[Change list 6](#_Toc294083068)

[How to Install .NET Bio Framework 9](#_Toc294083069)

[Prerequisites 9](#_Toc294083070)

[System Requirements 9](#_Toc294083071)

[Installation 9](#_Toc294083072)

[Migration to Newer Versions 11](#_Toc294083073)

[Installer Behavior 11](#_Toc294083074)

[DLL versioning 11](#_Toc294083075)

[The .NET Bio Framework Architecture 12](#_Toc294083076)

[.NET Bio Framework Components 14](#_Toc294083077)

[The Project Samples 16](#_Toc294083078)

[Resources 18](#_Toc294083079)

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.

© 2011 The Outercurve Foundation.

Distributed under Creative Commons Attribution 3.0 Unported License.

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 .NET Bio Framework, 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 Open Source Initiative (OSI).

## What is .NET Bio Framework

.NET Bio Framework 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

The Framework 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, .NET Bio Framework 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 site <http://bio.codeplex.com/> :

* Choose the downloads tab to install executables
* Choose the Source Code tab for source code
* Frequently Asked Questions, Training links, Sample Applications are all available on the home page

We encourage you to provide feedback on .NET Bio Framework at <http://bio.codeplex.com/discussions>.

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 bio.Silverlight library allows you to run under Silverlight which supports several mainstream OS platforms. There is a [Mono](http://monodevelop.com/) implementation of Silverlight called [Moonlight](http://www.mono-project.com/Moonlight) which is an open source implementation of Silverlight for instance. |

# Contribution Roles

There are two avenues of participation in the project:

* As a Contributor – downloading the deployed code base on Codeplex and submitting your work through CodePlex. You only need to register an account on CodePlex to be able to contribute.
* 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.

The following figure illustrates the two participation roles.



While contributors only have access to the periodically deployed 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://bio.codeplex.com/documentation>.

# Benefits of .NET Bio Framework

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

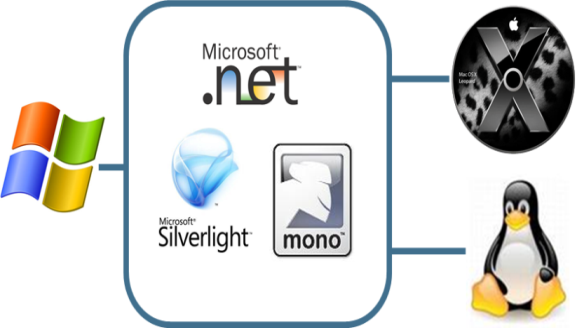
For these reasons, the need has been recognized for reusable libraries of high-quality bioinformatics code. The goal for the .NET Bio Framework is to provide a development framework for the bioinformatics community that ensures the high architectural and coding standards necessary for ease of application 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 the Framework 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](http://monodevelop.com/) or SharpDevelop.



**Note**: Mono is an open source implementation of Microsoft’s .NET Framework that runs on non-Windows operating systems. Silverlight is a 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 the Framework to perform a wide range of tasks, including:

* Assembly of Large Genomes: This is where the researcher would like to use the .NET Bio Framework (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 creating a sequence segment, reversing 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

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

# 

# What’s New and Changed

The .NET Bio Framework’s origins are from the Microsoft Biology Foundation (MBF) and Microsoft Biology Tools (MBT). The following features and tools were added, removed or changed in the migration from MBF to .NET Bio Framework.

## 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.

Change list for .NET Bio Framework 1.0

|  |  |
| --- | --- |
| Change | Description |
| AzureBlast | Removed. |
| Bio.Silverlight | Added. Bio.Silverlight is the .dll implementation of the Silverlight functionality for .NET Bio and allows application developers to develop cross platform applications utilizing the Silverlight methodology. |
| [Comparative Assembly](#_Comparative_Assembly) | New assembly techniques provided to allow for re-sequencing and comparative assembly of genomes to a reference of the same species or a similar species. |
| ComparativeUtil | **New** - A utility to kick off the comparative assembly. |
| Padena assembly algorithm | Increased capacity to support assembly of large genomes. Performance improvements to De Bruijn graph generation. |
| PadenaUtil | A tool allowing for command-line de novo assembly of genomic sequences. |
| Source tree changes | MBF\Source\MBF -> Bio\Source\Framework  MBF\Source\MBF - > Bio\Source\Tools |
| Wiggle format support | The Wiggle format is a genomics file format designed to display dense continuous data such as GC percent, probability scores, and transcriptome data. For reference go [here](http://genome.ucsc.edu/goldenPath/help/wiggle.html). |
| 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 | Any use of the term Fasta replaced with FastA to be consistent with the industry norm for this filetype. |
| 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 is named **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. |
| 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>.** |
| Data Virtualization | **Removed** |
|  |  |

**Note**: Documentation for utilities can be found in the same folder as the utility source code at ..\.NET Bio\Source\Tools.

# How to Install .NET Bio Framework

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

## Prerequisites

To use the capabilities of the Framework, 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 using Microsoft Visual Studio® to develop .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 .NET Bio Framework applications are described in the “Programming Guide.”

## Installation

The .NET Bio Framework project periodically posts stable snapshots of the source tree to Codeplex at <http://bio.codeplex.com/>. You can obtain a copy of the source tree by downloading a [snapshot](http://bio.codeplex.com/SourceControl/list/changesets).

If you are interested in .NET Bio Framework but do not want to submit code to the repository, you can just run the .NET Bio Framework installer, Bio.msi, and select the **Complete** install option to install the software development kit (SDK) provided on Codeplex. This option installs everything that you need to implement .NET Bio Framework applications, including all .NET Bio DLLs, under the $\Program Files\.NET Bio directory. However this option provides the project libraries and not the source code so you cannot modify the underlying source code.

Simply register on CodePlex to exercise either of these options. You do not need committer or contributor status for such downloads, the downloads are available to any interested user.

More details and the installer, .NET Bio.msi, are available at <http://bio.codeplex.com/releases>.

To install .NET Bio Framework

1. Download the .NET Bio installer, .NET Bio.msi, from <http://bio.codeplex.com/releases> to a folder on your hard drive (you can also download and run from the Codeplex site).
2. Navigate to that folder and double-click **.NET** Bio.msi, which runs the installation wizard.
3. Follow the directions in the installation wizard to install .NET Bio.

**Note**: Select **Complete** install if you want the source and binaries and that will include the SDK.

The .NET Bio Framework installer creates a directory named C:\Program Files (x86)\.NET Bio\1.0\Framework that contains the following:

\Add-ins

\Bio.Comparative.dll

\Bio.Padena.dll

\Bio.Pamsam.dll

Bio.dll

Bio.Silverlight.dll

Bio.WebServiceHandlers.dll

Readme.txt

If you install the optional SDK, the installer creates an SDK folder in the .NET Bio\1.0 folder that contains the following:

\SDK

\Docs

Bio.chm

Coding\_Conventions.docx

Commenting\_Conventions.docx

Comparative Assembly Technical Guide

Committer\_Guide

Contribution\_Documentation\_Template.docx

Contribution\_Guide.docx

Getting\_Started.docx

Becoming\_A\_Committer

Onboarding.docx

Overview.docx

PaDeNa Technical Users Guide.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 Bio.dll is installed.

\SDK

\Framework

\ComparativeUtil

\ConsensusUtil

\IronPython

\LayoutRefinementUtil

\LISUtil

\MumUtil

\NucmerUtil

\PadenaUtil

\ReadSimulator

\RepeatResolutionUtil

\SAMUtils

\ScaffoldUtil

\TridentWorkflows

# Migration to Newer Versions

For those that have a previous version of .NET Bio Framework installed you are not required to uninstall it in order to install a newer version. Versions can be installed side-by-side. Likewise if you are migrating from Microsoft Biology Foundation (MBF) to .NET Bio Framework.

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

The current Framework version  
v1.0

## 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).

## DLL versioning

The following dll versioning conventions are observed:

* .NET Bio DLL’s the current version 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.

# The .NET Bio Framework Architecture

The Framework 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.

BIO

Namespace

BIO.IO

Namespace

BIO.Web

Namespace

BIO.Algorithms

Namespace

.NET 4.0 Framework

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

The Framework 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.

Your

application

.NET Bio

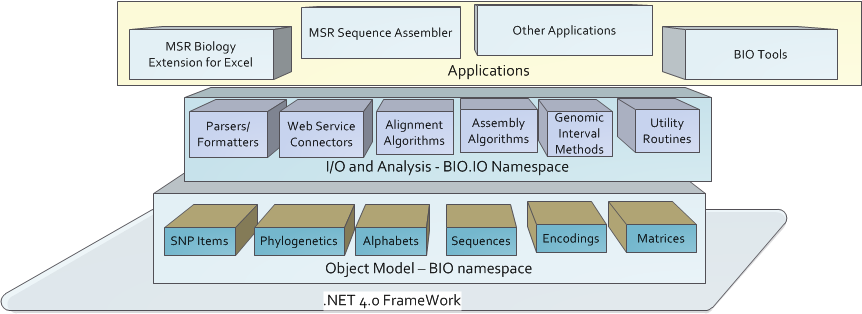
.NET 4.0 Framework

Your application/visualization

Using .NET Bio

Using .NET Bio Framework is as simple as adding a reference to Bio.dll to your project. You can then begin using the available types. Use bio.Silverlight.dll to develop Silverlight applications.

Sequences are the core concept in the Framework. 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.



## .NET Bio Framework Components

The Framework 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: .NET Bio Extension for Excel, an add-in for Microsoft Excel, and .NET Bio Sequence Assembler, a .NET application. For more information, see the accompanying documents at <http://bio.codeplex.com/documentation> or in the **..\Bio\Doc** document folder of the source code tree.

### Tools

The following tools are available for the deployed project:

Tools 2.0.Beta1

|  |  |
| --- | --- |
| Tool | Description |
| .NET Bio Sequence Assembler | Provides a packaged sequence assembler GUI.  .NET Bio\_Sequence\_Assembler\_User\_Guide.docx |
| .NET Bio Biology Extension for Excel | Provides an Excel toolbar ribbon to access .NET Bio Framework functionality.  .NET Bio\_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 |  |
| Wiggle | Parser and Formatter | Supports annotations. |
| 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 |
| EBI | ..\Bio\Source\Framework\Bio.WebServiceHandlers |
| NCBI | ..\Bio\Source\Framework\Bio.WebServiceHandlers |
| BLAST | Handler Bio.Web.Blast.IBlastServiceHandler at ..\Bio\Source\Framework\Bio\Web. |
| ClustalW | Handler Bio.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)\.NET Bio\1.0\Tools and the source code is located at $/BIO/SourceSamples. |
| BioDemo.py | An IronPython demonstration of some of the current non-GUI features.  By default deployed to C:\Program Files (x86)\.NET Bio\1.0\Tools and the source code is located at $/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)\.NET Bio\1.0\Tools and the source code is located at $ /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 .NET Bio Framework 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

.NET Bio Framework

* <http://bio.codeplex.com/>  
  .NET Bio Framework: Overview   
  .NET Bio Framework: Programming Guide   
  .NET Bio Sequence Assembler: User Guide  
  Padena: Parallel DeNovo Assembler
* Training Workshop Material - <http://mbf.codeplex.com/SourceControl/changeset/changes/76446>

.NET Bio Extension for Excel User’s Guide

<http://bio.codeplex.com/>  
.NET Bio 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/)