

# **NeuroEvolution Visualization Toolkit**

**Author: Derek James**

**Version 1.0  
May 4, 2004**

## Contents

1. Getting Started
2. Generating Graphics
  - 2.1 Generating Network SVG
  - 2.2 Generating Complexity SVG
  - 2.3 Generating Fitness SVG
  - 2.4 Generating Species SVG

# 1. Introduction

## 1.1 Rationale

There are a number of neuroevolutionary techniques currently being used in research, and though the particulars of their methodology, and the platforms and languages in which they are written vary widely, there all have in common the need for good visualization of data.

XML-based languages offer a strong, text-based, platform-independent, human-readable format for representing neural networks and information about their evolution. SVG, or scalable vector graphics, is an XML-based graphical language. Style sheets written in XSLT can easily and quickly convert XML documents into SVG, viewable with any browser, or with graphical programs that can handle the format.

The goals were to produce a suite of visualization tools with the following attributes:

- Built entirely on stable, open technologies
- Platform-independent
- Viewable with open technologies

The result is the NeuroEvolution Visualization Toolkit (NEVT), a package of text-based tools which will hopefully allow designers and users of different neuroevolutionary programs to quickly and easily represent their data visually.

## 1.2 What You Need

In order to produce graphical output using NEVT, you will need the following installed:

Java JRE:

<http://java.sun.com/j2se/1.4.2/download.html>

Xalan XSLT Processor

<http://xml.apache.org/xalan-j/downloads.html>

In order to view the SVG output in a standard browser, you will need one of the following plug-ins:

Adobe SVG Viewer

<http://www.adobe.com/svg/viewer/install/main.html>

Corel SVG Viewer

<http://www.smartgraphics.com/Downloads.shtml>

In order to edit/manipulate the SVG output, you may use one of the following free SVG editors:

Sodipodi

<http://www.sodipodi.com/index.php3?section=download>

Amaya

<http://www.w3.org/Amaya/User/BinDist.html>

## 1.3 Directory and File Structure

After installing the necessary software/plugin and downloading the NEVT .zip file, unzip the file to your preferred location (e.g., c:/nevt).

There are four directories, one for each type of visualization available:

- complexity
- fitness
- network
- species

Each directory contains:

- A sample XML file from which each type of graphical output is generated
- A DTD associated with the XML
- The XSL style sheet(s) used to generate the graphics
- A .bat file to run the processor from the command line
- A sample of SVG output

## 2. Generating Graphics

### 2.1 Generating Network SVG

In order to generate the network SVG, the XML format must adhere to the format of the network XML located in the Network directory. If you have a well-formed network XML document, follow the steps below to generate the resulting network SVG:

- 1) Make sure the XML document (e.g. "network34.xml") is located in the same directory as the DTD, and XSL style sheet(s).
- 2) From the command line, navigate to the Network directory.
- 3) From the command line, type "draw [filename]" (e.g. "draw network34").

Or...

If you are not using a DOS prompt, you may wish to write a shell script as a convenience. Otherwise, you may execute Xalan directly from the prompt using the following commands:

```
$ java org.apache.xalan.xslt.Process -in [filename].xml -xsl  
format.xsl -out formatted.xml
```

```
$ java org.apache.xalan.xslt.Process -in formatted.xml -xsl  
draw.xsl -out [filename].svg
```

- 4) The XML will then be processed into an SVG file, which will be generated in the same directory. To view the SVG file open it with a standard browser (making sure the SVG Viewer plug-in is installed) or one of the free SVG editors mentioned in the previous section.

## 2.2 Generating Complexity SVG

All other graphics in the NEVT are generating from the run XML.

In order to generate the complexity SVG, the XML format must adhere to the format of the run XML. Samples are provided in each of the Fitness, Complexity, and Species directories. If you have a well-formed run XML document, follow the steps below to generate the resulting SVG:

- 1) Make sure the XML document (e.g. "run77.xml") is located in the same directory as the DTD, and XSL style sheet(s).
- 2) From the command line, navigate to the Complexity directory.
- 3) From the command line, type "complexity [filename]" (e.g. "complexity run77").

Or...

If you are not using a DOS prompt, you may wish to write a shell script as a convenience. Otherwise, you may execute Xalan directly from the prompt using the following commands:

```
$ java org.apache.xalan.xslt.Process -in [filename].xml -xsl  
graphComplexity.xsl -out runComplexity.xml
```

- 4) The XML will then be processed into an SVG file, which will be generated in the same directory. To view the SVG file open it with a standard browser (making sure the SVG Viewer plug-in is installed) or one of the free SVG editors mentioned in the previous section.

## 2.3 Generating Fitness SVG

In order to generate the fitness SVG, the XML format must adhere to the format of the run XML. Samples are provided in each of the Fitness, Complexity, and Species directories. If you have a well-formed run XML document, follow the steps below to generate the resulting SVG:

- 1) Make sure the XML document (e.g. "run77.xml") is located in the same directory as the DTD, and XSL style sheet(s).
- 2) From the command line, navigate to the Fitness directory.
- 3) From the command line, type "fitness [filename]" (e.g. "fitness run77").

Or...

If you are not using a DOS prompt, you may wish to write a shell script as a convenience. Otherwise, you may execute Xalan directly from the prompt using the following commands:

```
$ java org.apache.xalan.xslt.Process -in [filename].xml -xsl  
graphFitness.xsl -out runFitness.xml
```

- 4) The XML will then be processed into an SVG file, which will be generated in the same directory. To view the SVG file open it with a standard browser (making sure the SVG Viewer plug-in is installed) or one of the free SVG editors mentioned in the previous section.

## 2.4 Generating Species SVG

In order to generate the species SVG, the XML format must adhere to the format of the run XML. Samples are provided in each of the Fitness, Complexity, and Species directories. If you have a well-formed run XML document, follow the steps below to generate the resulting SVG:

- 1) Make sure the XML document (e.g. "run77.xml") is located in the same directory as the DTD, and XSL style sheet(s).
- 2) From the command line, navigate to the Species directory.
- 3) From the command line, type "species [filename]" (e.g. "species run77").

Or...

If you are not using a DOS prompt, you may wish to write a shell script as a convenience. Otherwise, you may execute Xalan directly from the prompt using the following commands:

```
$ java org.apache.xalan.xslt.Process -in [filename].xml -xsl  
graphSpecies.xsl -out species.xml
```

- 4) The XML will then be processed into an SVG file, which will be generated in the same directory. To view the SVG file open it with a standard browser (making sure the SVG Viewer plug-in is installed) or one of the free SVG editors mentioned in the previous section.