

"Software that writes Software"
"Stochastic, Evolutionary, MultiRun Strategy Auto-Generation"

TRADING SYSTEM LAB
Product Description
Version 1.1
05/28/10

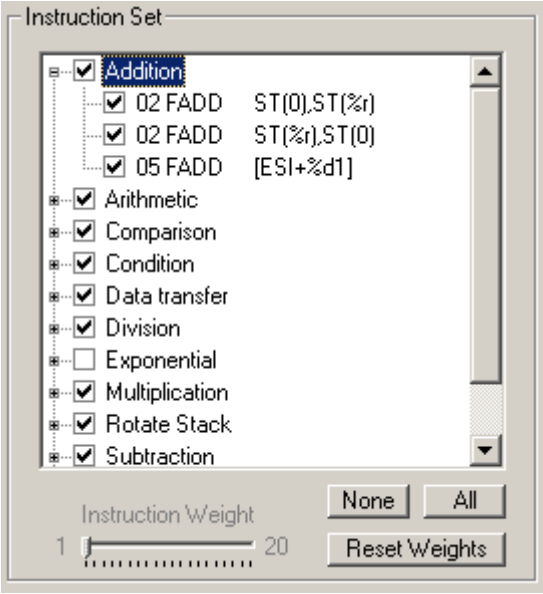
Trading System Lab (TSL) will automatically generate various Trading Strategies in as little as a few minutes using a very advanced computer program known as an AIMGP (Automatic Induction of Machine Code with Genetic Programming). Register level genetic manipulations and an advanced trading simulator provides for convergence to solutions far faster than other algorithms. TSL is a significant departure from the current traditional manual design workflow used to create trading strategies. No trading strategy exists prior to the initiation of a run in TSL as is the case in traditional in Genetic Algorithm optimization for example. Trading strategy logic, triggers, inputs and equations are written for you in what has been described as an explosion of mathematics. Single Market Directional Trading Systems, Pairs Trading Systems, Portfolios Trading Systems, Daytrading Systems and Options Trading Systems may be automatically evolved with TSL. Creation of a Trading System within Trading System Lab is accomplished in 3 easy steps. First, a simple preprocessor is run using your development platform or TradeStation® (or any of our other data preprocessors) that automatically extracts and preprocesses the necessary data from the market you wish to work with. Second, the Trading System Lab is run for about 2 to 20 minutes (Intel Core 2 Duo or equivalent, 10-20 years daily data) to evolve a new Trading System. Third, the evolved Trading System is translated and exported into C, JAVA, Assembler, Easy Language®, C# or several other languages to produce new Trading System signals.

Trading Systems that took weeks or months to produce in the past may now be produced in minutes through the use of Trading System Lab. Trading System Lab is a platform for the automatic generation of Trading Systems. TSL makes use of a high speed Genetic Programming Engine and will produce Trading Systems at rates faster than 2 Million evolved Trading Systems per hour (on fast processors such as the Intel Core i7 980x). **Note that we are not simply running a brute force optimization, nor are we using older technology such as a Genetic Algorithm, looking for optimum parameters from which to use in an already structured Trading System.**

With this application you will be able to automatically create Trading Systems far faster than conventional methods involving manually writing Trading System code. Trading System Lab makes use of a Linear, Automatic Induction of Machine Code with Genetic Programming Algorithm under an exclusive license from Register Machine Learning, Inc (RML). This Genetic Programming engine has been extensively studied and has been applied to many fields of and applications and is currently being studied by the U.S. military. For additional information please view and download appropriate white papers at http://www.tradingsystemlab.com/whitepaper.aspx, including white papers on speed and accuracy studies from Register Machine Learning, Inc., Jet Propulsion Lab (JPL) and Scientific Applications International Corporation (SAIC).

Trading System Lab not only evolves Trading Systems extremely fast, but it also writes Trading System code. Additionally, the evolutionary algorithm begins the design of the Trading System from a zero point origin. In other words, at the beginning of a TSL run, no Trading System exists, and no Trading System code exists. The only Trading System elements that exist at the start of a run are the initial genetic material, known as a terminal set, which would be known by traders as "patterns, indicators and data". A maximum of 56 inputs may be used as this starting point. This terminal set consists of preprocessed facts in either a Boolean (0 or 1) or a Numeric (finite numbers) form. Only basic Trading System and Genetic Programming constraints are defined at the beginning of the run. The trader or Trading System developer may use his own data, filters, indicators, inter-market data, fundamental data or patterns within TSL. Additionally, the developer may inject his own "domain expertise" in feeding this initial genetic material to TSL. This way the intelligence gathered over time from the trader may be taken advantage of by TSL during the course of the evolutionary path to your genetically engineered Trading System.

TSL then combines this initial genetic material with 34 instruction sets such as simple addition, subtraction, multiplication and division, or more complex stack rotation, conditionals or data transfers and randomizations, producing new Trading System equations in terms of machine instructions operating directly within the registers of your CPU. For example, there are several ways to perform an addition in terms of machine code:



The Genetic Program will use these various machine code instructions to write or rewrite Trading System equations.

The general concept here is “Machine Learning applied to Trading System design”.

In a recent study only a filter bank was used as the initial genetic material. TSL then automatically created Trading Systems in a just a few minutes that displayed adequate Out of Sample performance. In another study, only 9 simple candlestick patterns were used as the initial genetic material. TSL then evolved Trading systems in a few minutes that also displayed adequate Out of Sample performance.

TSL may also enhance, reengineer, modify or expand upon any existing Trading System or Technical Analysis approach by adding the existing Trading Systems or Technical Analysis facts, equations, preprocessing, or logic to TSL’s internal development logic. To accomplish this, the user simply merges signals derived from the existing Trading System with additional preprocessed facts. The TSL platform will then take this mixture of facts, rearrange and combine them with the GP’s function set so as to create a new Trading System. Simple Parameter Optimization would not suffice, nor does it take place, however a new Trading System structure emerges, perhaps with characteristics derived from both the legacy Trading System and the newer preprocessed facts.

TSL begins at a zero point origin making no assumptions about the movement of the market in the future and then "evolves" Trading Systems at a very high rate combining information present in the market and formulating new filters, functions, conditions and relationships as it progresses towards a "genetically engineered" Trading System, then writes the portable code for you.

It is important to note that mathematical equations are written for you at the completion of the evolutionary path in the form of code in a variety of languages.

Key evolutionary issues that make this approach the best approach for Trading System development are:

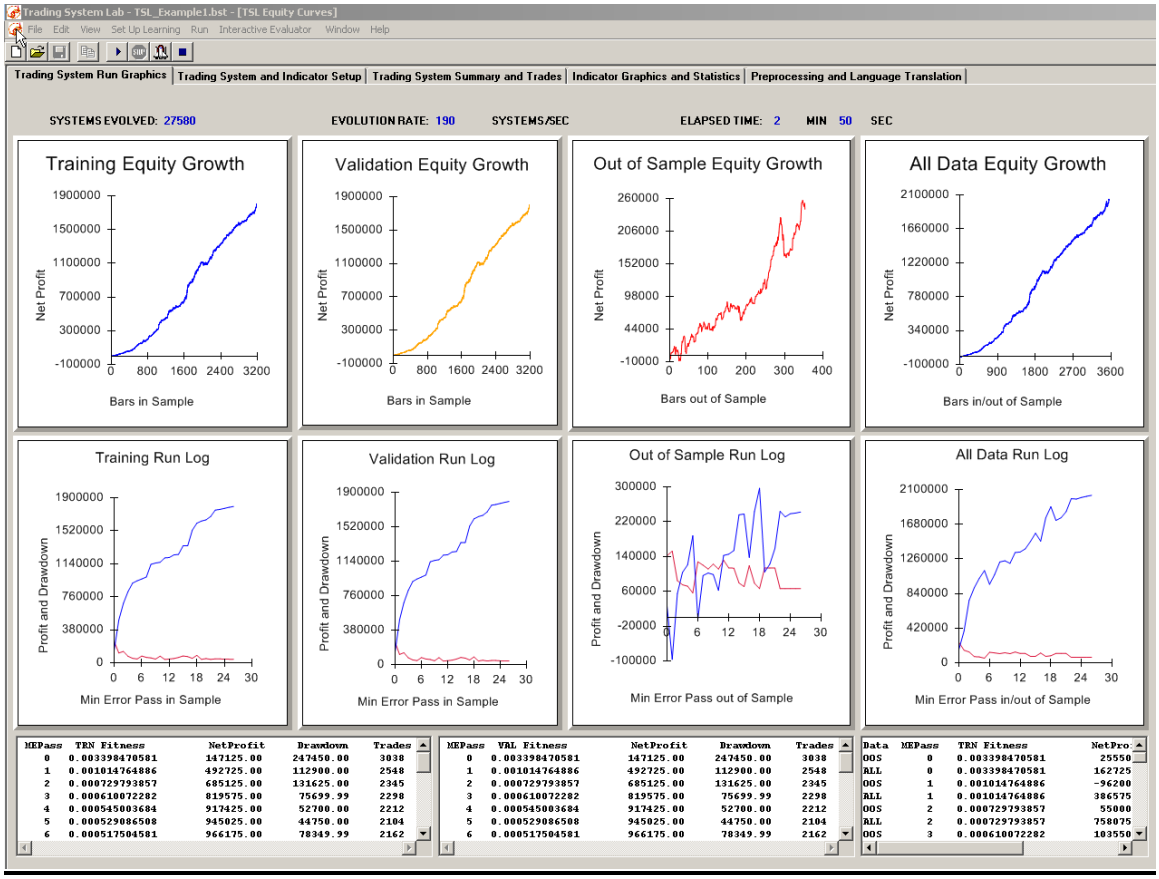
1. Rapid convergence at rates up to 200 times faster than conventional methods. (See speed and accuracy white papers from JPL and SAIC)
2. Input data set reduction occurring far faster than conventional methods. (Often only 5 to 10 parameters are used, out of the initial 56 inputs, in the final Trading System resulting in a simple, efficient Trading System)
3. Out of Sample results displayed both **during** the evolutionary process as well as **immediately available upon completion.** (Allows "path" intelligence to be seen.)
4. Efficient model complexity reduction through Parsimony Pressure. (Draws from minimal description length concept; Simpler systems are better.)
5. Non-homologous crossover resulting in very dynamic evolutions. (Varies the size of the evolved code. This is a critical LAIMGP concept.)
6. Production of C-code, Java, Assembler, EasyLanguage® and others. (Allows for portability and integration into industry API's, platforms, and Algorithmic Trading Engines.)

The result is that an excellent Trading System may be generated in a few minutes using 20-30 years of market data on virtually any market.

It is important to note that Genetic Programs are superior to the older and less efficient technology known as Genetic Algorithms. Further discussion on this topic may be found on the web site and in demos.

The TSL platform is the result of the convergence of many languages and technologies into an operational whole. Since TSL will either create a Trading System or indicate rapidly if a market is too random to allow the creation of a Trading System, we believe that TSL will soon become the standard for development of new Trading Systems and a metric for performance measurements of existing Trading Systems. Since TSL's speed grows with the rate of processor speed improvements, faster processors will allow yet greater and greater functionality and convergence speed. We estimate that within a few years, TSL will be producing Trading Systems in seconds. We are investigating concurrent, parallel, and multi core/multi processor configurations. Our plans include the capabilities to design Trading Systems on most of the major securities and commodity markets in the world quickly and efficiently.

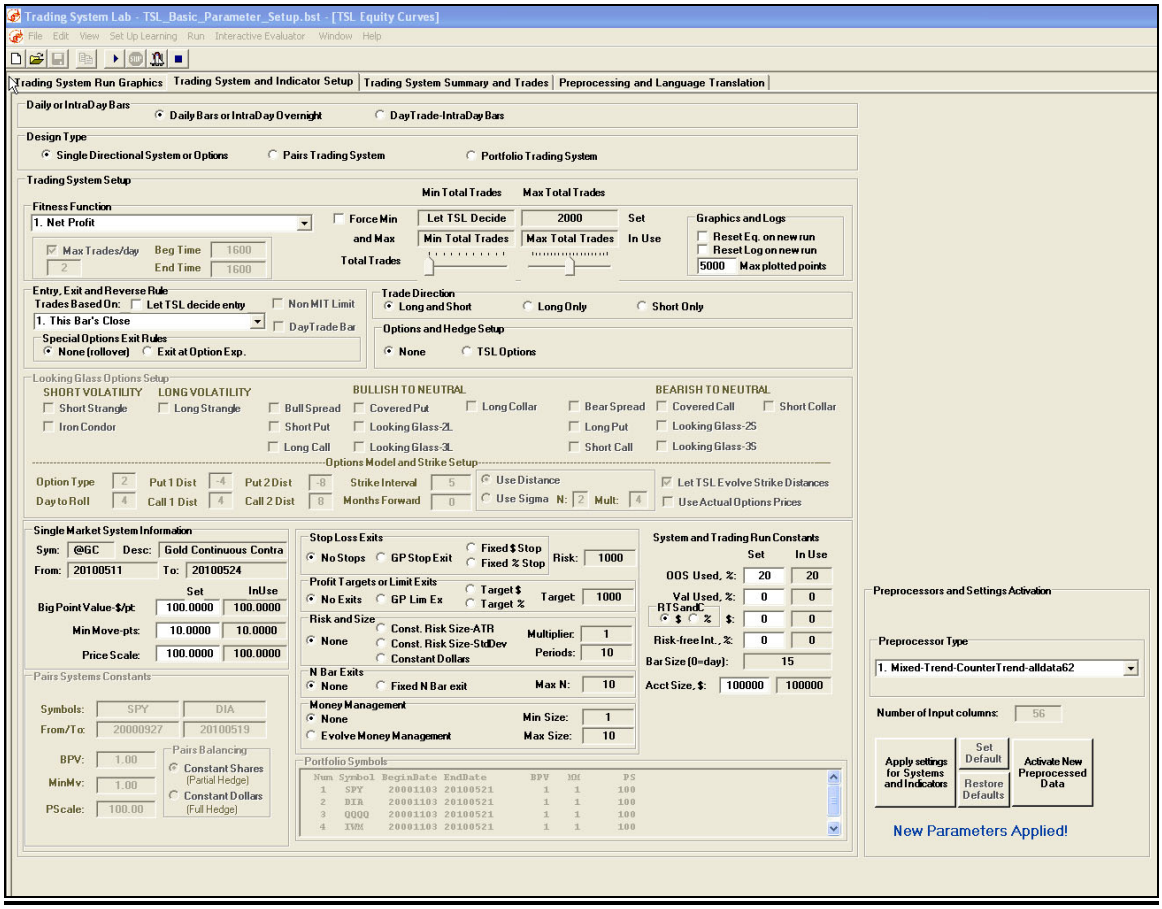
Equity Graphics Tab:



There are 8 graphs and 3 numeric readouts on the Run Graphics Tab. In the upper row, Training, Validation, Out of Sample and the All Data equity curves are displayed. These graphs show the profit or loss over time of the most recent system that best meets the requirements of the selected fitness function. Numeric run log information is also displayed assisting the developer in choosing the best trading model.

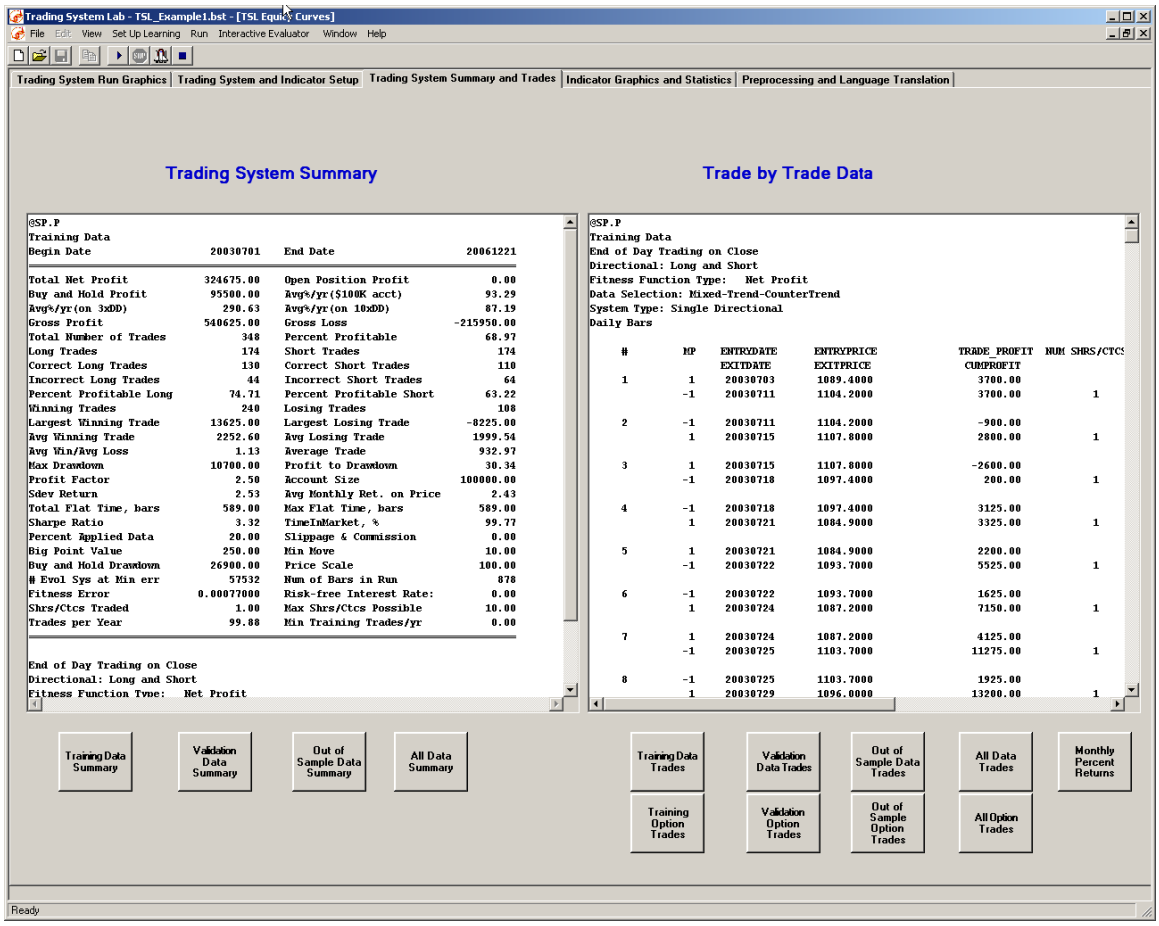
Out of Sample test data is data that was not used by the GP during evolution. It is important to note that OOS data is not part of the Training process and is considered blind, hidden, applied, held out or walk-forward data. Testing on Out of Sample (OOS) data is critical in the determination of the degree of robustness of a Trading System. **TSL displays both the latest performance of our Trading System on the OOS data and the OOS performance of our Trading System "during" the evolutionary process.** If the general path of the OOS log shows continual poor performance during the stochastic path, not implementing that particular configuration, data set, or market-model combination should be considered.

System Setup Tab:



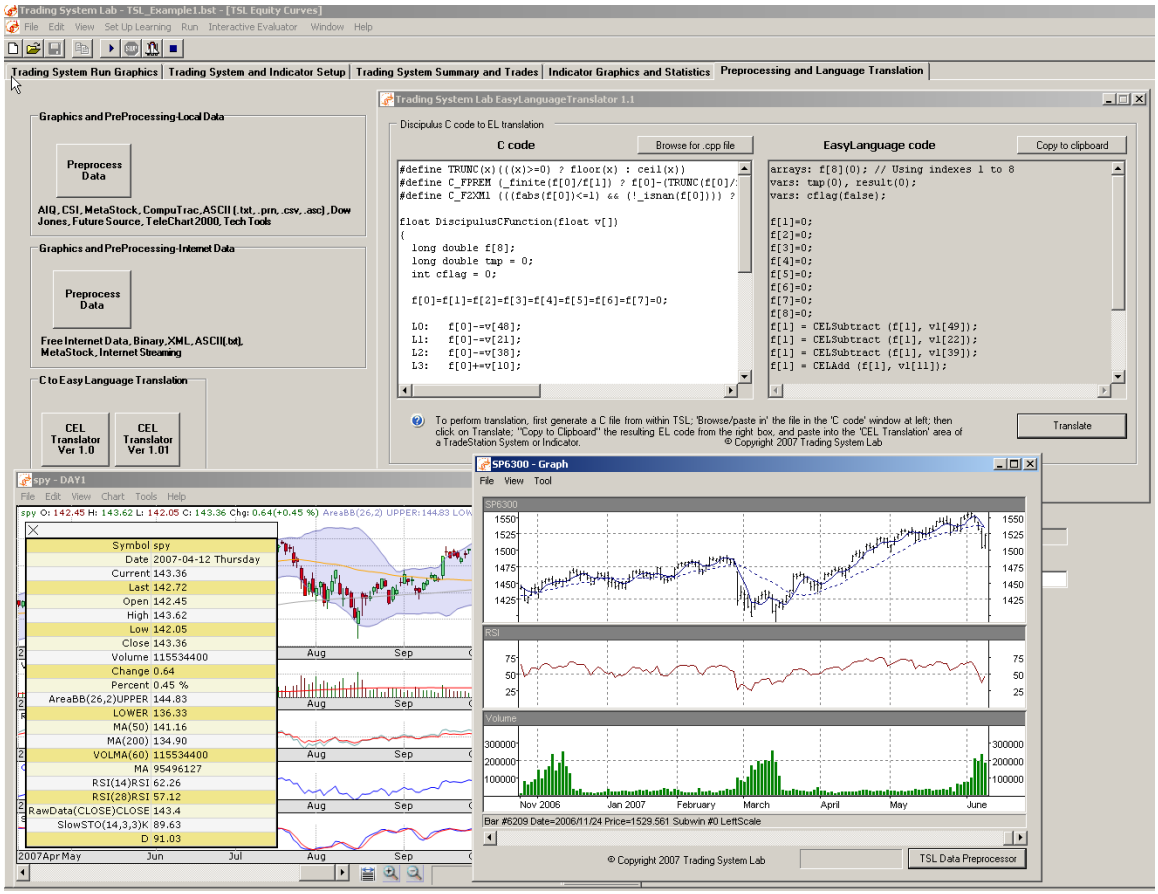
The Trading System Setup Tap allows the configuration of 33 Fitness Functions, 19 evolved Entry Tactics, 5 Preprocessed Data Sets, 7 evolved Exit Tactics and numerous other Trading System design configurations. TSL has ongoing and frequent updates and enhancements with Version 1.1 being a major update following many earlier updates and numerous client and trader feedback.

System Results Tab:



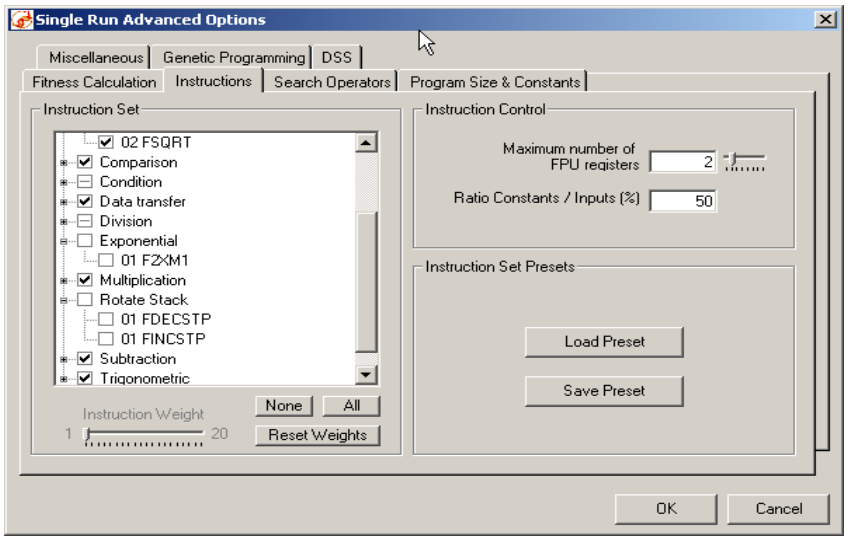
The Results Tab shows the Trading System Summary and the Trades during the run as well as when the run is finished. All information is held in spreadsheet importable text files. The Options trade window shows over 70 Options Trading Systems columns. System Summary information is available for Single Market Directional Trading Systems, Pairs, Portfolios and Options Trading Systems.

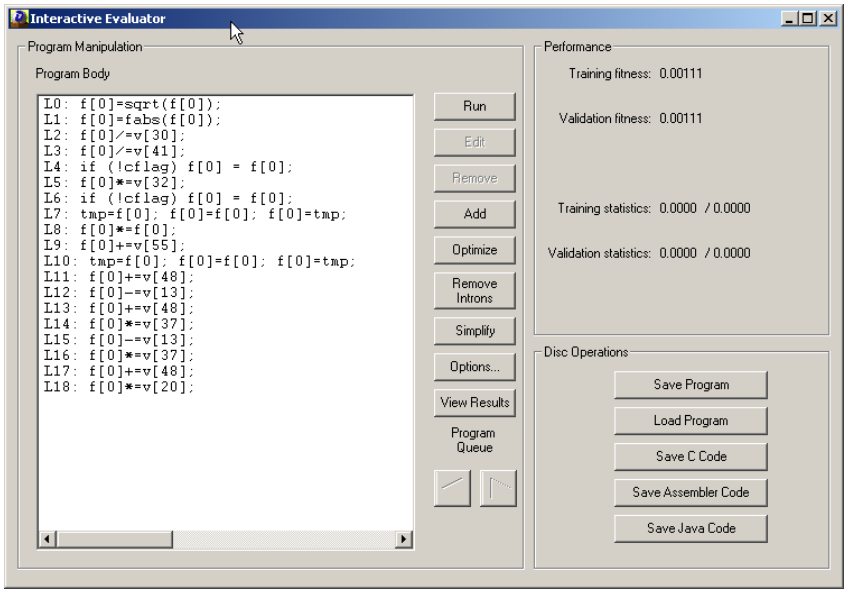
Language Translation and Preprocessing Tab:



The Preprocessing and Language Tab provides for data preprocessing, C to EasyLanguage® translation, options database preprocessing (if available), and TSL platform license settings. The functionality here is extremely simple providing for fast preprocessing and language translation. Available data formats include: CSI, MetaStock, AIQ, TradeStation®, Free Internet data, ASCII, TXT, CSV, CompuTrac, DowJones, FutureSource, TeleChart2000v3, TechTools, XML, Binary and Internet Streaming data. Available endpoint languages include C, JAVA, Assembly, EasyLanguage® and others. Check with TSL for the latest information on currently available languages.

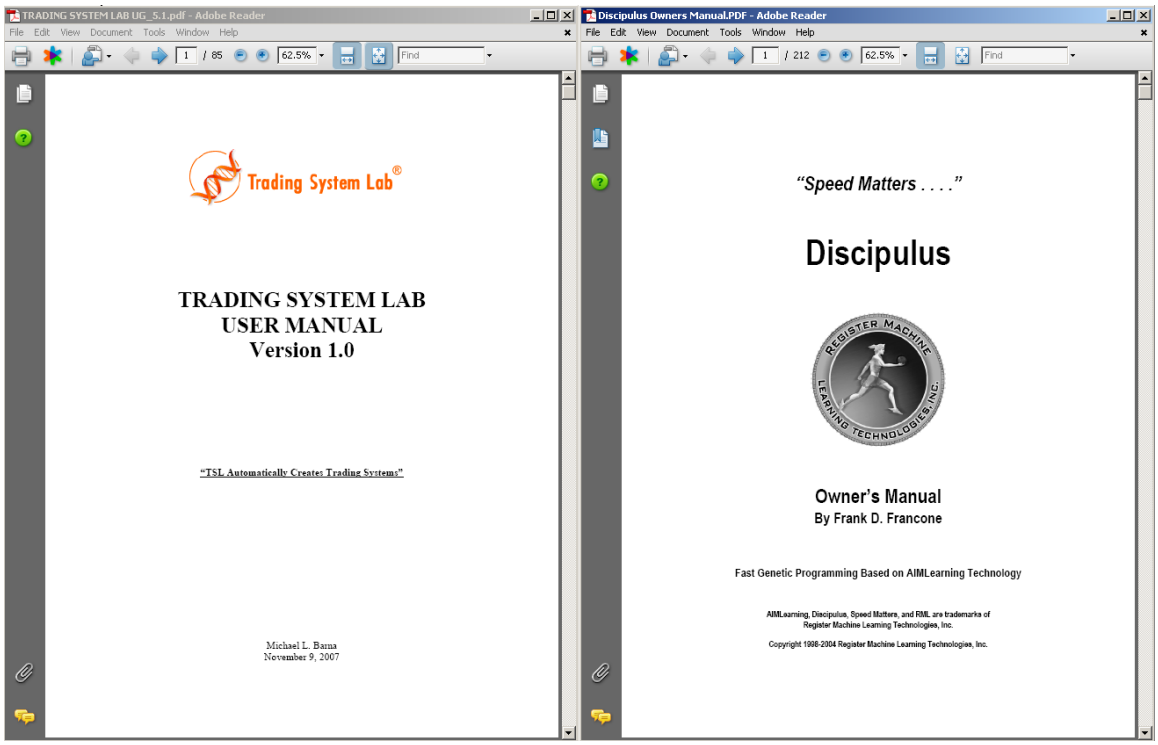
Genetic Programming Setup Section:





GP instruction sets may be edited, or weights adjusted for each machine instruction. The GP produced C code may be edited and viewed in the Interactive Evaluator window following evolution, prior to translation to other languages The Genetic Program used within TSL is hundreds of times faster than competing algorithms. Moreover, TSL writes the Trading System code for you.

TSL Platform User Manuals:



The Trading System Lab and Discipulus Genetic Programming User Manuals provide nearly 300 pages of support. Discussions on how to develop robust trading systems are included as well as methods to enhance OOS performance. Technical Support is available directly through TSL.

For pricing information, please visit our web site.

Copyright 2010 Trading System Lab

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement. The software may be used and copied only in accordance with the terms of those agreements. No part of the publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Trading System Lab.

Trading System Lab® is a trademark of Trading System Lab, San Jose, California

AimLearning, Discipulus, Speed Matters and RML are trademarks of Register Machine Learning Technologies, Inc., Littleton, Colorado.

Trading System Lab
7227 Glenview Drive
San Jose, CA
408-356-1800
www.tradingsystemlab.com

Register Machine Learning Technologies, Inc.
7606 S. Newland St.
Littleton, CO. 80128
www.aimlearning.com