Evolución de la Interfaz de Consulta de la *SfGD*. Un Puente de Entendimiento Informático-Biológico

Alfonso Pons[1], Claudia Reynares[1], Laura Angelone [1,2], Paula Fernández[3], Pilar Bulacio[1,2], Norma Paniego[3], Elizabeth Tapia[1,2]

[1]*Facultad de Cs. Exactas e Ingeniería, Av. Pellegrini 250, Rosario, Argentina*
[2]*CIFASIS-Conicet, Bv. 27 de Febrero 210 Bis, Rosario, Argentina*
[3]*Instituto de Biotecnología, CICVyA, INTA Castelar, Las Cabañas y Los Reseros, (B1712WAA) Castelar, Provincia de Buenos Aires, Argentina*

Resumen

La base de datos SfGD (Sunflower Genomic Database) fue diseñada para almacenar, gestionar y consultar información genómica de ESTs de girasol dentro del marco de un proyecto llevado a cabo en INTA Castelar. Inicialmente los datos biológicos fueron registrados en una base de datos Access, recurso válido pero acotado por el enorme volumen de datos biológicos derivados de los estudios actuales. La ventaja de una base de datos radica en almacenar grandes volúmenes de datos en forma organizada. El biólogo expresa los resultados de su investigación "en un lenguaje" "con terminología específica" desconocida o poco conocida por los informáticos no avezados en la Biología. El informático logra abstraerse de la realidad para desarrollar un programa en un lenguaje sintáctica y semánticamente diferente al que maneja el biólogo. Esto genera un distanciamiento entre ambos que afecta a la aplicación desarrollada. Sin embargo, el informático debe interiorizarse de la dinámica de los datos biológicos para poder construir un gestor que a modo de puente permita unir ambas ciencias. En este trabajo presentamos la evolución de la interfaz que se utilizó para consultas en la base de datos SfGD, arribando a una interfaz que permite consultar la base en un lenguaje coloquial y menos técnico, sabiendo que esta herramienta será, en un futuro cercano, un recurso de acceso público para la comunidad científico/académica interesada en consultar información molecular asociada a esta especie de interés agronómico.

Palabras Clave
Base de Datos, comunicación informático-biólogo, girasol, EST.

Abstract
The SfGD(Sunflower Genomic Database) was designed to store, management and consult sunflower ESTs genomic information, within the framework of a project carried out at INTA Castelar. At the beginning the biological data was stored in an Access database, a valid but restricted resource due to the enormous amount of data coming from current research. The advantage of a database lies on the capacity to store large amounts of data in an organized way. The biologists express the results of their investigations in "a language with specific terminology" unknown for the informatics. The informatics manages to abstract way to develop a program with a language syntactically and semantically different from the one used by the biologist. This situation generates a gap between both professionals that affects the developed application. However, the informatics must embrace the biological data dynamics in order to build a program that works as a bridge between both sciences. In this work we introduce the evolution of the interface utilized to consult the SfGD. Arriving to an interface that allows consulting in a more colloquial and less technical language, knowing that this tool will be, in a close future, a public access resource for the scientific/academic community interested in consulting molecular information related to this species of agronomical interest.

Key words
Database, interface informatics-biologic, sunflower, EST.

1. Introducción

La base de datos SfGD (Sunflower Genomic Database) [1] fue diseñada para almacenar datos e información proveniente de la ejecución de proyectos de genómica funcional [2]. En particular, el conjunto de datos incluye secuencias de ESTs y unigenes derivados del ensamblado de los mismos, su anotación funcional [3] y datos asociados a respuestas funcionales ligados a estrés abiótico (frío y salinidad) en plantas de girasol (*Helianthus annuus L.*)[4]. Los datos en SfGD han sido compilados con el fin de encontrar funcionalidades génicas ligadas al conjunto de genes representados por los mencionados ESTs. Brevemente, las secuencias ESTs son comparadas, ensambladas y agrupadas para definir un conjunto no redundantes de genes. Finalmente estos genes son comparados con secuencias de proteínas descriptas y disponibles en bases de datos públicas con el fin de asignar funciones moleculares.

La necesidad de almacenar digitalmente el enorme volumen de datos biológicos que resultan de esta investigación, y las limitaciones impuestas por Access tanto en la compatibilidad con otras herramientas similares de uso libre como en la publicación y acceso a los datos en forma pública y remota, hizo que se requiriera a un grupo de profesionales de la Informática (CIFASIS-FCEIA) la creación de una base de datos (BD) que permita al usuario la carga, organización, y visualización gráfica de los datos generados en el proyecto EST de girasol mediante una interfaz web.

La elección de las herramientas de software a utilizar fue sencilla. La amplia disponibilidad de ellas se redujo considerablemente al fijar las siguientes premisas para la selección:

1. Software libre.
2. Portabilidad
3. Independencia del sistema operativo
4. Escalabilidad
5. Arquitectura cliente servidor
6. Humanizar la interfaz
7. Afinidad con alguna de ellas o conocimiento previo de las mismas.
8. Aprovechar los recursos de Internet

Inicialmente, y como resultado del análisis de estas premisas, se decide implementar una base de datos relacional utilizando el paquete de software LAMP (Linux, Apache, MySQL y Programación (PHP + SQL)) [5] [6]. MySQL para la creación de la SfGD, PHP para la programación de los clásicos formularios de ABM (Alta, Baja y Modificación), Apache como servidor web y Linux como sistema perativo soporte de los anteriores softwares .

El primer paso para crear esta BD fue el trabajo conjunto entre los biólogos y los informáticos para tomar en consideración los datos resultantes de los trabajos de investigación. Con esta información se creó e implementó una primera estructura con el objetivo claro de optimizar el almacenamiento de datos y efectivizar su gestión. Las bases de datos relacionales se basan en conceptos de orden y estructuras. En ellas la información se almacena como estructuras comúnmente denominadas tablas. Cada tabla almacena información de una única entidad y se establecen reglas que vinculan o relacionan la información entre las distintas entidades.

La esencia de una base de datos relacional y gran parte de su poder proviene del adecuado diseño de las tablas y sus relaciones entre sí.

Para evitar problemas asociados a la redundancia, y a la actualización de los datos en las tablas y con el fin de proteger la integridad de los mismos, se normalizó la BD de acuerdo a la 3FN (Tercera Forma Normal) [7][8].

Una BD tiene la ventaja de almacenar grandes volúmenes de datos de forma estructurada, lo que disminuye el espacio ocupado digitalmente, debido a la eliminación de datos redundantes producto de la duplicación de información. Por otra parte la elección de una adecuada estructura estandarizada permite la obtención de información valiosa que surge de exhaustivas búsquedas y de comparaciones entre los datos (cruce de información). El gestor de BD es una herramienta útil para llevar a cabo tareas administrativas, tanto sobre la estructura de la base como sobre los datos en ella almacenados. El desarrollo de sistemas Gestores de BD con interfaces específicas permite a los programadores enmascarar la base, delimitar responsabilidades, y otorgar permisos de acceso a las diferentes calidades de usuarios.

En el proceso de implementación de un sistema informático, los desarrolladores se logran abstraer de la realidad para construir un programa escrito en un lenguaje sintáctica y semánticamente diferente al que manejan los biólogos. Por otra parte, éstos últimos expresan los resultados de sus investigaciones "en un lenguaje" "con terminología específica" desconocida o poco conocida por los informáticos. Este inconveniente afecta a las aplicaciones desarrolladas por los programadores, que ponen esfuerzos en dar la mejor solución desde su propio punto de vista. Como consecuencia de esto se genera un distanciamiento entre ambos grupos que afecta a la aplicación desarrollada. En este punto, los informáticos deben interiorizarse de la dinámica de los datos biológicos para poder construir aplicaciones (en este caso un gestor de bases de datos) que a modo de puentes permitan unir ambas ciencias. Deben aprender, de a poco y con ayuda, los aspectos importantes de la Biología vinculados al proyecto.

De lo expresado se plantean dos desafíos:
1) crear una base de datos con un modelo sencillo y eficiente,
2) diseñar un nivel de visualización sencillo y análogo al lenguaje coloquial y/o visual de comunicación.

El nivel de visualización, es lo que los usuarios finales puede visualizar del sistema, describe sólo una parte de la base de datos, en este nivel no está presente la estructura de la BD, es decir, las tablas y las relaciones entre ellas.

Uno de los objetivos principales de los sistemas de bases de datos es proporcionar a los usuarios finales una visión abstracta de los datos, y esto se logra enmascarando ciertos detalles técnicos-informáticos de almacenamiento y mantenimiento de los datos.

Considerando que el sistema puede proporcionar muchas visiones para la misma base de datos, se deberá encontrar la más sencilla y adecuada para los biólogos.

Una primera aproximación fue implementar una lista con vínculos a las consultas rutinarias realizadas por el grupo de investigación del INTA Castelar (Fig.1). Sin embargo, las consultas debieron ser reformuladas periódicamente pues, esta forma de implementación es estática y necesita de la re-programación por parte de los programadores, es decir, establece una fuerte dependencia con el informático.

El segundo intento por mejorar esta interfaz de consulta consistió en elaborar formularios (clásicos de toda aplicación web) pero utilizados de forma no convencional, con el objetivo

de agilizar y hacer más intuitiva la consulta (Fig.3-a y Fig. 3-b). Esta solución, si bien es independiente del informático sigue siendo fuertemente dependiente de la estructura de la BD.

El tercer intento por establecer un nivel de visión simplificado y análogo al lenguaje coloquial, tal vez no el último, fue pre-establecer frases que se puedan interconectar para armar oraciones y realizar la consulta, tratando de lograr la independencia del informático y ocultando la estructura de la BD.

2. La construcción del puente

Para el análisis de datos la solución más rápida y efectiva que se puede aplicar en un modelo de BD relacional es utilizar el lenguaje de consultas estructurado SQL. Este lenguaje permite obtener información precisa, clasificada y ordenada de la base. La calidad de la información que se logre recuperar de la base está condicionada fuertemente al conocimiento de la estructura interna de la misma y al dominio de la sintaxis del lenguaje SQL, tanto cuanto más compleja sea la estructura de la base, y cuanto más específica sea la búsqueda o consulta. Tal vez la consulta expresada en lenguaje coloquial sea tan simple como querer saber "¿Cuántos… hay…?" o "¿Cuáles cumplen con la siguiente regla…?", pero expresarla en SQL implica conocer el modelo de datos con buena precisión y en el caso de los biólogos solicitar ayuda a los informáticos (Fig. 1). Por lo cual es necesario recurrir a algún tipo de interfaz de usuario para presentar, comparar y analizar los datos.

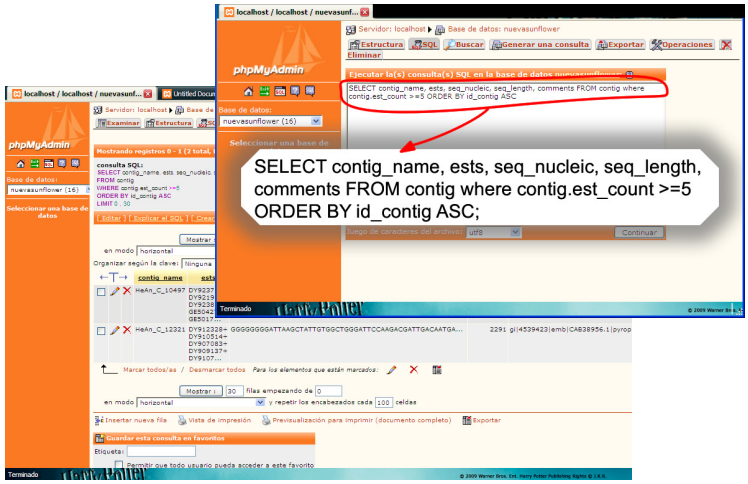

Fig. 1 Interfaz clásica de una consulta en lenguaje SQL

Una interfaz web es conveniente porque es un ambiente familiar, relativamente fácil de implementar y accesibles simultáneamente por varios usuarios.

A continuación se describe como evolucionó la interfaz de consultas de la SfGD desde sus inicios.

2.1 Una interfaz para consultas rutinarias

Como primer intento se pre-establecieron consultas rutinarias que surgieron del trabajo habitual de los investigadores del INTA Castelar (Fig.2). Se automatizaron para que puedan ejecutarse con un solo *click* tantas veces como sea requerido.
Esta forma de trabajo fue una primera solución rápida y efectiva tanto desde la perspectiva informática como biológica. Sin embargo, le quitaba dinamismo a la aplicación y seguía siendo totalmente dependiente de los programadores. Cada vez que los biólogos querían obtener nueva información debían solicitar a los informáticos la re-programación de las consultas.

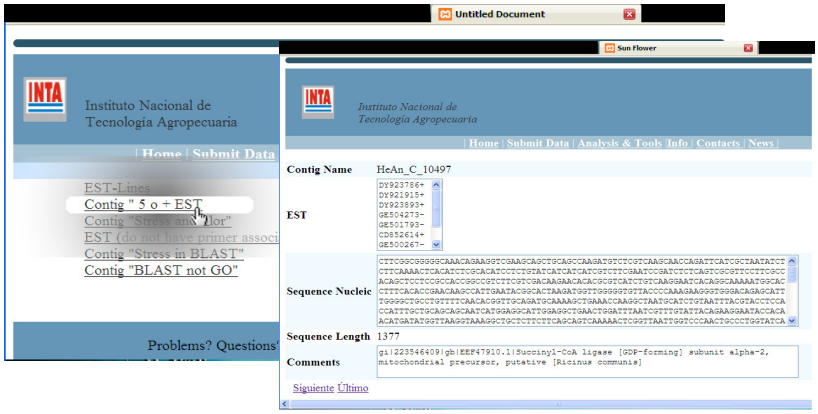

Fig. 2 Consulta rutinaria: "*mostrar todos los contig con 5 o más EST*"

2.2 Una interfaz gráfica avanzada

Un escenario habitual en las aplicaciones de BD para manipular los datos es un formulario con una interfaz gráfica. Generalmente los programadores simplifican la interfaz, implementándola como un cuadro de texto acompañado de un botón BUSCAR que examina la base en busca de coincidencias, sin posibilidad alguna de especificar donde realizar la búsqueda o que tipo de resultados mostrar.

En muchos casos la búsqueda se realiza sobre una única tabla de la estructura de la base, consecuentemente los informes que resultan no establecen ningún tipo de relación entre los datos almacenados en diferentes tablas, esto es, no se cruza la información.

Por ello es que nuestro desarrollo se dirige a una interfaz que muestre una estructura simplificada e intuitiva de la información almacenada en la base. La misma permite seleccionar los datos que se desea exhibir, pudiendo filtrar cada uno de ellos estableciendo criterios de comparación (mayor que, menor que, igual a) o de coincidencias (contiene, comienza con, termina con).

En este tipo de interfaz, los usuarios pueden solicitar ver la información sin importar como se encuentran almacenados los datos, y en caso de ser necesario el programa será el encargado de cruzar la información. (Fig. 3-a y Fig. 3-b)

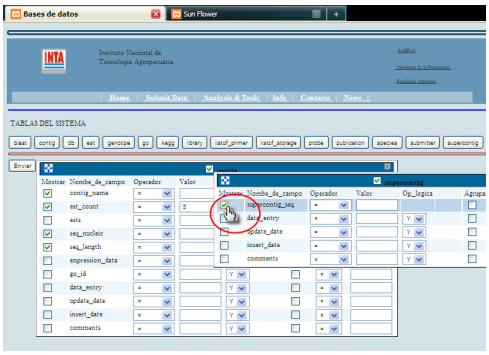

Fig 3-a Interfaz de consulta visual de la SfGD

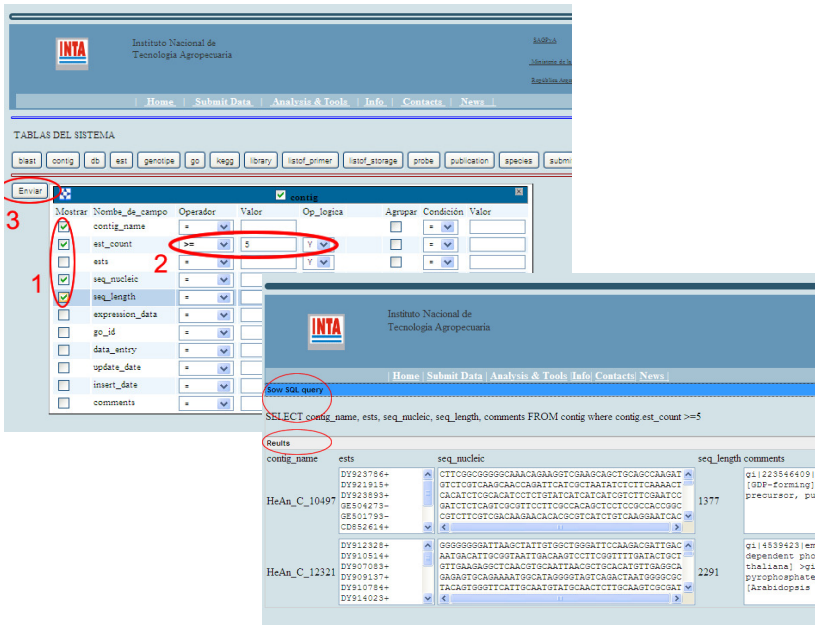

Fig 3-b Consulta por formulario: "*mostrar todos los contig con 5 o más EST*"

2.3 Una interfaz coloquial

Como tercer intento, tal vez no el último, para optimizar un nivel de visión sencillo, posible y análogo al lenguaje coloquial fue pre-establecer frases que se puedan interconectar para armar oraciones y realizar la consulta. Logrando la independencia del informático y enmascarando la estructura de la BD.

Para crear las consultas en lenguaje coloquial se emplearon herramientas habitualmente aplicadas en formularios pero utilizadas de forma no convencional.

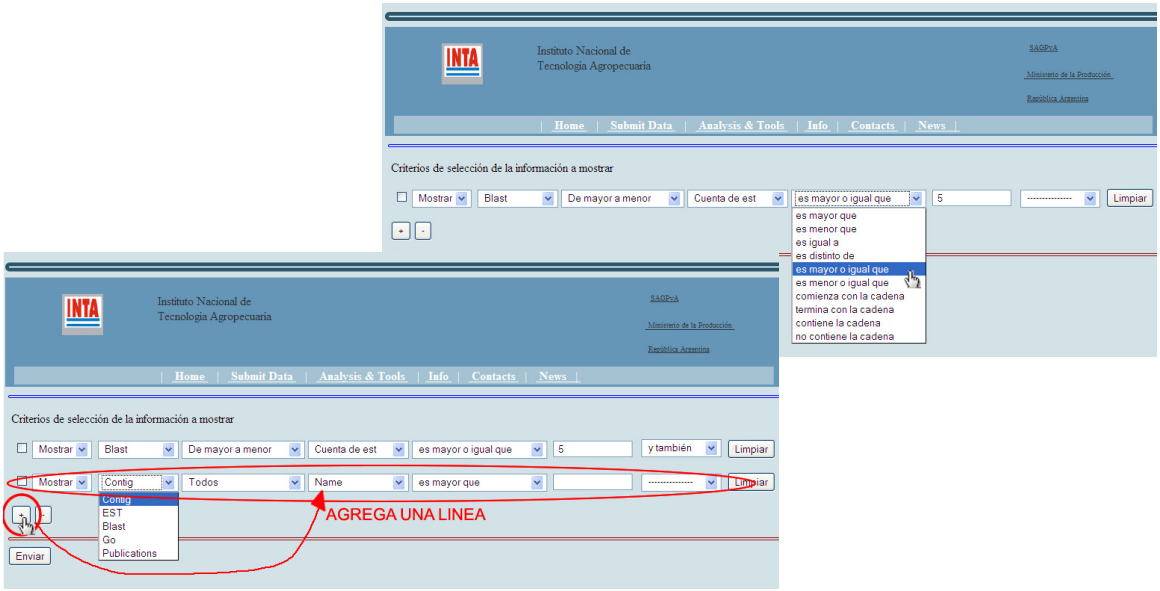

Fig. 4 La misma consulta de la Fig. 3-b se reduce a seleccionar en una línea las opciones adecuadas. Se pueden agregar más condiciones en nuevas líneas.

La interfáz consiste en seleccionar frases del lenguaje coloquial que se exhiben en listas desplegables, enmascarando así las instrucciones del lenguaje SQL. Estas frases se pueden combinar, acotando los resultados, estableciendo órdenes y jerarquías. Si los resultados obtenidos son satisfactorios se puede guardar la consulta realizada para hacer que el proceso sea reversible y recuperarla para repetir la búsqueda modificando o no alguno o todos los parámetros o simplemente para repetir la consulta con nuevos datos. (Fig. 4)

El desafío actual es construir una interfaz que funcione como un puente de comunicación, con la idea de ser una especie de traductor sintáctico y semántico para que los usuarios (en este caso los biólogos) puedan escribir o seleccionar por medio del mouse, en forma gráfica, que tipo de información necesitan obtener de la base de datos sin necesidad de involucrarse en los aspectos técnicos relacionados con la administración de la misma.

Para responder a la premisa de humanizar la interfaz, principalmente en lo que a obtención de información se refiere, se recurrió a la utilización de código jQuery [9]. Este es un framework (conjunto de bibliotecas javascript), que permite simplificar la manera de interactuar con los documentos HTML. Es software libre y de código abierto que posee un doble licenciamiento: MIT [10] y GNU [11].

Lograr este objetivo requirió de permanentes consultas a las fuentes biológicas para comprender y corregir los programas en pos de mejoras que atiendan a su usabilidad.

3. Conclusión

Este trabajo está encuadrado en el desarrollo de software multidisciplinario y colaborativo [12]. Por lo cual se desea que la solución de construir un sistema de consulta con interfaz web en un lenguaje coloquial y menos técnico, haciendo uso de herramientas simples como el mouse, un browser y elementos visuales, sea un puente que ayude a reducir la brecha informática-biólogica. Una interfaz sencilla permitirá obtener resultados fructuosos para la comunidad científica interesada en consultar información molecular asociada a esta especie de interés agronómico.

Debe notarse que una interfaz amigable reduce los tiempos necesarios para la selección de aquellas hipótesis de investigación más prometedora, y favorece la formulación de nuevas hipótesis.
También se pretende que la aplicación perdure, crezca y evolucione en el tiempo en función de las necesidades y nuevos requerimientos de los profesionales que la utilicen en sus trabajos de investigación.
Puede concluirse que una interfaz amigable puede ser un elemento importante o critico para un proyecto de investigación en Bioinformática.

Referencias

[1] Fernández P., Angelone L., Bulacio P., Reynares C., Tapia E., Paniego N. (2009). SfGD: Base de Datos Genómicos de Girasol, Congreso de Agro informática 2009, CAI 2009, 24 al 28 de agosto 2009, ISSN 1852-4850.
[2] Fernandez P, Paniego N, Lew S, Hopp HE, Heinz R (2003). Differential representation of sunflower ESTs in enriched organ-specific cDNA libraries in a small scale sequencing project. BMC Genomics 2003, 4:40.
[3] Lew, S., Fernández, P. y Paniego, N, eBiopipeline: una plataforma abierta para el procesamiento de datos bioinformáticas, Segundas Jornadas Argentinas de Agroinformática (JAIIO), 9-10 de setiembre de 2008, SantaFe, Argentina.
[4] Fernández P, Di Rienzo J, Fernandez L, Hopp HE, Paniego N, Heinz RA. Transcriptomic identification of candidate genes involved in sunflower responses to chilling and salt stresses based on cDNA microarray analysis. BMC Plant Biol. 2008; 8(1):11.
[5] Lee, James; Brent Ware (December 2002). Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP. Addison Wesley. ISBN 0-201-77061-X
[6] Conrad Bessant, Ian Shadforth and Darren Oakley. "Building Bioinformatics Solutionswith Perl, R and MySQL", 2009. Oxford University Press. ISBN13: 9780199230235 - ISBN10: 0199230234
[7] Silberschatz A., Korth H., Sudarshan S, *Fundamentos de bases de datos*, 2006, Ed. McGRAW-HILL, ISBN 9788448146443.
[8] Peter Rob / Carlos Coronel, *Sistemas de bases de datos,* 2004, Cap. 4, Ed. Thomson, ISBN 970-686-286-2
[9] http://docs.jquery.com/Main_Page
[10] http://www.opensource.org/licenses/mit-license.php
[11] GNU General Public License. http://www.gnu.org/licenses/
[12] Bassi,S., Gonzalez, V., Parisi, G. "Computational Biology in Argentina", PLoS Computational Biology, 2007

Datos de Contacto *Laura Angelone. CIFASIS-Conicet, 27 de Febrero 210 bis, Rosario, 2000, Argentina. angelone@cifasisconicet.gov.ar.*