# A pipeline design for downloading and analyzing promoter sequences in *Solanum lycopersicum*

Alejandro D. Pistilli[1], Guillermo R. Pratta[2, 3], Laura Angelone[4, 5] and Debora P. Arce[2, 6]

[1]*Facultad de Ciencias Agrarias, Universidad Nacional de Rosario (UNR).*
[2]*Instituto de Investigaciones en Ciencias Agrarias de Rosario, CONICET, UNR.*
[3]*Cátedra de Genética, Facultad de Ciencias Agrarias. UNR.*
[4]*Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas, Rosario.*
[5]*Facultad de Ciencias Exactas, Ingeniería y Agrimensura, UNR.*
[6]*Grupo de Análisis, Desarrollos e Investigaciones Biomédicas,*
*Facultad Regional San Nicolás, Universidad Tecnológica Nacional.*

***Abstract*— A pipeline architecture is implemented to automatize gene promoter sequence download from tomato genome Solanum lycopersicum annotated in Sol Genomics Network. Output gene promoters can be analyzed with MEME and TOMTOM programs. The code is available at www.github.com/lalebot/pip-prom-tom and Git is used as control versions software. Combined Python threads, regular expressions, and SQLite databases are used to reduce time for downloading sequences and optimize informatic resources. The methodology presented in this work is potentially applicable to other biological fields.**

***Keywords*— Bioinformatics, Plant Biology Systems, Gene expression analysis, MEME, TOMTOM, Phyton, SQLite, threads.**

***Resumen*— Se presenta el desarrollo de una arquitectura en pipeline que automatiza la descarga de promotores de *Solanum lycopersicum* desde la Sol Genomics Network y luego los analiza con los programas MEME y TOMTOM. El código está disponible en www.github.com/lalebot/pip-prom-tom y utiliza Git como software de control de versiones. Se combina el uso de *threads* en Python, expresiones regulares y base de datos SQLite para que conjuntamente disminuyan el tiempo de descarga de los promotores y optimicen la utilización de recursos informáticos. La metodología que presenta este trabajo es potencialmente aplicable a otras áreas biológicas.**

***Palabras clave*— *Bioinformática, Biología de Sistemas en Plantas, Análisis de la expresión génica, MEME, TOMTOM, Phyton.***

## I. Introducción

Las nuevas tecnologías de secuenciación han permitido la generación de grandes cantidades de información que está siendo recopilada en bases de datos y en diferentes plataformas bioinformáticas. Esta información está disponible para la comunidad científica y consta de secuencias génicas, estructura y localización de genes, herramientas para visualización y manipulación de secuencias de ADN, ARN y proteínas. Además, contienen resultados de experimentos de transcriptómica, tales como secuenciación del ARNm (RNA-Seq) y microarreglos.

Las bases de datos y plataformas más relevantes para las solanáceas (tomate, papa, tabaco, entre otras) y específicamente para el tomate, son actualmente: Sol Genomics Network (SGN, https://solgenomics.net), Tomato Functional Genomics Database (http://ted.bti.cornell.edu) [1], Tomato Genomic Resources Database (http://59.163.192.91/tomato2/) [2]. La plataforma SGN es un repositorio primario de datos fenotípicos, genéticos, genómicos, de expresión génica y metabólica proveniente de la familia de las solanáceas y otras especies relacionadas. SGN almacena los datos de secuenciación del genoma de la variedad Heinz 1706 de *S. lycopersicum* como así también los de otras variedades y cultivares silvestres (ej. *S. pennelli* LA0716, *S. pimpinellifolium* LA1789). En esta plataforma reside una gran variedad de herramientas bioinformáticas que permiten la manipulación de estos datasets [3], posibilitando así que los genomas secuenciados sirvan de referencia para el estudio de otras variedades para las cuales aún no hay datos de secuenciación de nueva generación.

Esta información ha comenzado a ser utilizada para el mejoramiento de la calidad de los frutos de tomate en variedades locales de nuestro país [4]. En estas aplicaciones, suelen presentarse diversas limitaciones para el usuario en el manejo de las herramientas disponibles en este tipo de plataformas.

La conversión desde el estado de madurez del fruto de tomate (EMT) verde al estado completamente maduro rojo implica cambios dramáticos en el color, composición, aroma, sabor y textura del fruto. La maduración es un proceso que incluye alteraciones en el metabolismo y la expresión de genes, teniendo un efecto dramático en la calidad de los frutos. Los diversos EMT como así también la respuesta al estrés o diferentes situaciones fisiológicas vegetales, implican a nivel molecular, una reprogramación y una modificación de la regulación de la expresión de genes o grupos de genes específicos [5]. Los mecanismos implicados en la regulación de la expresión génica abarcan la interacción de una proteína o factor de transcripción (del

Contacto: Alejandro D. Pistilli. Facultad de Ciencias Agrarias UNR. Campo Experimental Villarino, Zavalla. Phone +543414970080 int 1122. *apistill@unr.edu.ar*

inglés, transcription factor, TF) con una secuencia corta (5-15 pares de bases o pb) o motivo de ADN.

Los motivos de ADN se encuentran en los promotores génicos y su identificación ha constituido por años un tópico de discusión en el ámbito científico. Un promotor es la región del genoma cercana al sitio de inicio de la transcripción (SIT) de un gen y generalmente se la ubica 1000 pb río arriba (upstream). Algunos autores describen aproximadamente 200 río abajo (downstream) del SIT [6].

La interacción entre los TFs y sus motivos en el ADN es específica y lleva a la inducción o represión de la expresión génica [7] [8]. Numerosos métodos han sido utilizados para identificar motivos en secuencias promotoras. Su identificación ha sido uno de los problemas más ampliamente estudiados, no sólo por su significado biológico sino también por su dificultad bioinformática. Esta dificultad deriva de la cantidad enorme de datos con los que se cuentan en la actualidad y de la necesidad de acceso a los mismos de formas flexibles y parametrizables para su análisis. Frecuentemente, es necesario analizar grupos de promotores provenientes de genes co-regulados, es decir aquellos genes que poseen un perfil de expresión similar, ya sea por inducción o represión en la expresión génica. La disponibilidad de herramientas bioinformáticas para el análisis de promotores dentro de las Solanáceas es baja, con ausencia de interfaces gráficas adecuadas para analizar grupos de genes simultáneamente. Tal es así que este trabajo se originó a partir de necesidades o preguntas biológicas por resolver, tal como se observa en dos Trabajos Finales de Especialización en Bioinformática UNR [9] [10]. En estos trabajos se contempló la necesidad de analizar promotores de genes de interés para la calidad de frutos de tomate y durazno, buscando motivos sobre-representados o bien construyendo bases de datos con motivos de interés.

En SGN se encuentra disponible una herramienta para la descarga de promotores con ciertas limitaciones, como la imposibilidad de obtener promotores de varios genes en forma simultánea. Si bien existe la posibilidad de descargar en bulk las zonas promotoras de todos los genes de S. lycopersicum cv Heinz 1706, estas zonas sólo poseen una cantidad de pb upstream/downstream del SIT prefijadas por SGN (https://solgenomics.net/tools/bulk?mode=ftp) y no permiten la flexibilidad en la selección de los parámetros asociados.

Por tal motivo, el presente trabajo tiene como objetivo principal desarrollar una herramienta que se conecte con la web de SGN, para la identificación, extracción y posterior análisis de secuencias de grupos de promotores con los programas MEME [11] [12] y TOMTOM [13]. MEME utiliza un algoritmo que identifica uno o más motivos en una colección de secuencias de ADN o proteínas. El archivo de salida de MEME es matricial y puede ser comparado contra bases de datos matriciales correspondientes a motivos conocidos, por ejemplo, la Jaspar DNA CORE (2016) [14] [15] [16]. Posteriormente, mediante el uso del algoritmo TOMTOM, se cuantifica estadísticamente la similitud entre el motivo incógnita o query obtenido por MEME y el motivo anotado en la base de datos de motivos conocidos. Así, MEME identifica motivos sobrerrepresentados en una lista de secuencias promotoras y TOMTOM clasifica estos motivos de ADN según sea su unión con diferentes familias de TFs específicos.

## II. Materiales y Métodos

La arquitectura en pipeline permite ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales, siendo la entrada de cada una la salida de la anterior.

Se desarrolló un *script* con las órdenes de procesamiento en el lenguaje de programación Python versión 3 (http://www.python.org) que funciona sólo bajo sistemas operativos GNU/Linux y derivados. Un *script* contiene órdenes que el programa intérprete lee una a una y las ejecuta secuencialmente. Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

El seguimiento y versionado del proyecto se realizó con Git (http://git-scm.com/). Git es un software de control de versiones diseñado por Linus Torvalds, uno de los creadores del sistema operativo GNU/Linux.

El proyecto está alojado en la web GitHub (https://github.com/lalebot/pip-prom-tom) y es de libre acceso. En la misma se permite su clonación, duplicación y posterior modificación por cualquier usuario de la web.

La descarga del proyecto contiene los siguientes archivos:

-*README.md,* es el manual de instalación y uso que incluye: requisitos para la ejecución, un instructivo para la descarga, detalle de los parámetros del archivo de configuración y detalle de los comandos que el *script* necesita para ejecutarse.

-El archivo de configuración inicial *conf.ini*

-*exa_prom.txt* que contiene una lista de códigos de promotores de ejemplo.

-*pip_prom_tom.py,* el código del *script.*

## III. Resultados

Se anexa a esta presentación un video tutorial on-line (https://youtu.be/QA1AEsjHLgU) donde se explica y visualiza paso a paso una ejecución del *script* de la arquitectura en pipeline.

Las entradas que requiere la arquitectua en pipeline son:

-Un archivo de texto con la lista de los códigos de promotores de *S. lycopersicum* a descargar.

-Un archivo de configuración con los parámetros personalizables para la ejecución del *script* y para la ejecución de los algoritmos MEME y TOMTOM.

Al ejecutar el *script*, se debe señalar el nombre del archivo que contiene la lista de promotores, el nombre que se le va a dar a ese proyecto, si van a descargarse los promotores upstream o downstream, el gap y si el *script* se ejecuta en modo pipeline o no.

En el presente trabajo nos referimos como gap a la cantidad de pares de bases extras que se descargarán contabilizando las bases a partir del SIT de cada gen. Si el gen está en la hebra (+), se suman las bases extra al SIT, y si el gen se encuentra en la hebra (-), estas bases extras se restan (Figura 1).

### *A. Procesamiento del pipeline*

-A partir de un archivo con la lista de códigos de promotores de SGN (Figura 2), el *script* comienza a hacer consultas para obtener los promotores a través de *threads* (hilos), que trabajan en paralelo para la optimización del rendimiento y ancho de banda de internet. La cantidad de *threads* se define en el archivo de configuración inicial

*conf.ini*. Este archivo puede modificarse antes de cada ejecución del *script*.

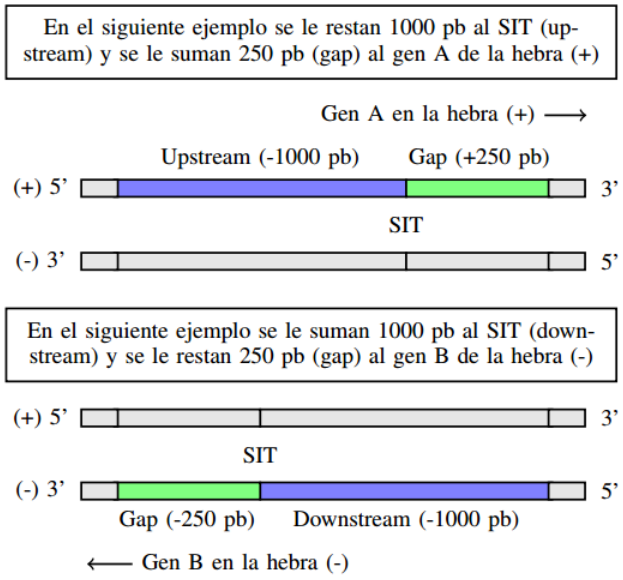


Fig 1: Obtención de la secuencia promotora, contando a partir del SIT para los genes A (con región codificante CDS en hebra positiva) y B (con CDS en hebra negativa).

-Cada uno de esos hilos o procesos hijos, descarga el código HTML del resultado de una búsqueda en SGN. Dentro de ese HTML, el *script* realiza la búsqueda de un código único que identifica al promotor dentro de SGN para realizar otra consulta y descarga el código HTML que devuelve esta última. Las búsquedas dentro de los códigos HTML se realizan con expresiones regulares.

-Una vez que el *script* tiene el código de identificación del promotor, realiza otra consulta para obtener el HTML en donde encuentra el SIT que sirve de referencia para hacer la descarga del promotor.

-Con el SIT encontrado se realiza la descarga de los pares de bases de ese promotor en base a los parámetros downstream/upstream y gap.

-El resultado obtenido del promotor se guarda en una base de datos SQLite que contiene todos los datos descargados y procesados. En el caso de que el proceso de descarga tenga una interrupción, el *script* puede retomar las búsquedas a partir de los datos incompletos de esta base de datos. La base de datos permite que todos los hilos estén leyendo y escribiendo a la vez sin que se pierda ningún dato ya que la misma gestiona el acceso simultáneo de varios procesos.

-Cada vez que el hilo encuentra y descarga correctamente un promotor realiza una nueva consulta a la base de datos para continuar la búsqueda de otro promotor.

-Una vez que los hilos han cargado toda la base de datos completando las cadenas de promotores faltantes, el *script* genera un archivo FASTA que contiene todas las cadenas de promotores.

-Este archivo sirve de entrada para que ser analizado por el programa MEME. Los parámetros de análisis que utiliza MEME son tomados del archivo *conf.ini*.

-El *script* descarga la última versión de base de datos de motivos que necesita el programa TOMTOM para realizar la comparación.

-Los resultados del algoritmo MEME sirven de entrada al programa TOMTOM que los compara con la base de datos de motivos de ADN conocida. La base con la cual se hace la comparación está establecida en el archivo *conf.ini*. Por defecto es la base "Jaspar DNA CORE (2014)" pero puede ser modificada con otras bases de datos [16].

-Si un error sucede durante el procedimiento, el mismo queda registrado en un archivo de nombre *logs.log*

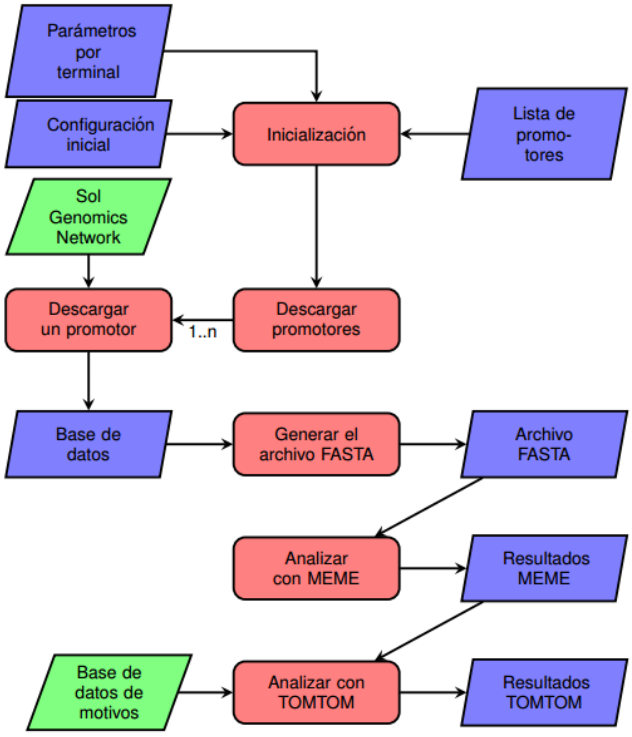


Fig 2: Diagrama de flujo de la arquitectura en pipeline. En rojo se muestran las etapas de la arquitectura en pipeline, en azul los archivos que se consultan y se generan y en verde las consultas a la web.

La arquitectura en pipeline está desarrollada con herramientas de código abierto:

-GNU/Linux OS: Sistema operativo.

-Python: Lenguaje de *scripting.*

-Git: Sistema de versionado.

-SQLite: Base de datos.

### *B. Impacto*

La arquitectura en pipeline brinda mayor flexibilidad en el uso de herramientas para la extracción y el análisis de secuencias de promotores en *S. lycopersicum* cv. Heinz 1706. El uso de esta arquitectura en pipeline puede extenderse a otras especies de solanáceas o variedades de tomate secuenciados. Para el caso de aquellos genomas que se hallan secuenciados de manera incompleta o parcialmente (genomas borrador) bastaría con modificar ciertas características del *script* para adaptarlo a sus nuevos usos. Por otro lado, y en relación al área biológica en la que impacta, la identificación de los motivos de ADN para la unión de TFs en familias de genes vinculadas con el proceso de maduración del fruto de tomate u otras situaciones fisiológicas, del desarrollo o bien vinculadas con la respuesta al estrés abiótico o biótico, permiten avanzar en el área de mejoramiento de la calidad y sabor de los frutos de tomate.

Esta presentación propone una modalidad de trabajo que puede ser replicada en otras áreas de desarrollo de herramientas bioinformáticas en donde se necesite una mecanización de descargas y análisis de datos en base a una lista de búsqueda.

### *C. Ventajas*

Utiliza Git, el mismo permite obtener la última versión de un proyecto para su clonación y posterior modificación. Explora la creación de un *script* de arquitectura en pipeline, ya que, al utilizar el sistema de control de versionado Git, no sólo está disponible el proyecto en su versión final sino en todas y en cada una de las etapas del mismo desde sus inicios. Se almacena en GitHub, una de las plataformas de desarrollo colaborativo más importantes de la actualidad. Se aplican conceptos de escalabilidad para brindar flexibilidad y adaptación a cambios.

Tanto el lenguaje de programación, el sistema operativo y el sistema de versionado son de código abierto y de libre uso.

El proyecto está publicado bajo la licencia GNU General Public License (GPL) 3.0, la cual exige la publicación del código fuente y que todos los trabajos derivados del original conserven la misma licencia GPL, no permite enlaces con módulos privativos (de código cerrado) y requiere que todos los cambios realizados a la versión original sean reflejados en el código fuente con sus respectivos autores.

El *script* utiliza *threads* para realizar las búsquedas de los promotores. Los *threads* son hilos de ejecución o subprocesos de un proceso padre que permiten mejorar el rendimiento. Se utilizan en procesos de *hacking* y en análisis heurísticos. Gracias a esta automatización en paralelo el resultado se obtiene cerca de diez veces más rápido que si no se usaran los mismos.

Se usan expresiones regulares para el análisis de los códigos HTML y la descarga de la base de datos de motivos que utiliza el TOMTOM.

## IV. CONCLUSIONES

Como resultado del presente trabajo, fue posible:

-Desarrollar una arquitectura en pipeline, en el lenguaje Python, que permita la automatización de la descarga de promotores de *S. lycopersicum* desde SGN para su posterior análisis con MEME y TOMTOM.

-La creación del proyecto en una plataforma de versionado de software (www.github.com/lalebot/pip-prom-tom). Git permite guardar cada paso del desarrollo del proyecto. Dentro del mismo se encuentran los manuales de instalación y uso.

-Implementación conjunta de *threads* en Python, expresiones regulares y base de datos SQLite para mejorar el rendimiento disminuyendo diez veces el tiempo necesario para una ejecución.

-Adaptar e implementar una metodología que es aplicable a cualquier área biológica siempre que haya un genoma anotado y disponible en la web.

-Ampliar las funcionalidades de la SGN al permitir la descarga personalizada de un número de pb río arriba/abajo conjuntamente con un gap, la interfaz web de la SGN no permite esto.

## TRABAJOS FUTUROS

Está previsto validar esta arquitectura de pipelines en otras especies de interés agronómico para las que existen bases de datos similares a Sol Genomics Network.

## AGRADECIMIENTOS

Este trabajo se realizó como parte de los requisitos del Ing. Alejandro D. Pistilli para obtener el Grado Académico de Especialista en Bioinformática de la Universidad Nacional de Rosario, quien agradece a la Comisión Académica y al Ministerio de Ciencia, Tecnología e Innovación Productiva de la Provincia de Santa Fe por las becas otorgadas para cursar dicha Especialidad.

A la Universidad Tecnológica Nacional (FRSN-UTN), ya que este trabajo está financiado por el PID-3938.

## REFERENCIAS

[1] Z. Fei, J. G. Joung, X. Tang, Y. Zheng, M. Huang, J. M. Lee, … J. J. Giovannoni, Tomato Functional Genomics Database: a comprehensive resource and analysis package for tomato functional genomics}, Nucleic Acids Research 39 (Database issue), D1156–D1163, http://doi.org/10.1093/nar/gkq991, 2011.

[2] B. V. Suresh, R. Roy, K. Sahu, G. Misra, D. Chattopadhyay, Tomato Genomic Resources Database: An Integrated Repository of Useful Tomato Genomic Information for Basic and Applied Research, PLoS ONE 9(1): e86387. doi:10.1371/journal.pone.0086387, 2014.

[3] The Tomato Genome Consortium, The tomato genome sequence provides insights into fleshy fruit evolution, Nature 485, 635–641, doi:10.1038/nature11119, 2012.

[4] V. Cambiaso, J. H. Pereira da Costa, G. R. Rodríguez, G. R. Pratta, L. A. Picardi, D. M. Francis, R. Zorzoli. Polimorfismo en la secuencia genómica completa entre un cultivar argentino y una especie silvestre de tomate (Solanum SPP.), Cátedra de Genética, Facultad de Ciencias Agrarias, Universidad Nacional de Rosario, Zavalla, Santa Fe, Argentina, XLIV Congreso Argentino de Genética -13 al 16 de septiembre de 2015- Mar del Plata, Buenos Aires, Argentina, 2015.

[5] D. Gierson, A. A. Kader. Fruit ripening and quality, The Tomato Crop: A scientific basis for improvement, Cap 6, 241-280, doi:10.1007/978-94-009-3137-4_6, 1986.

[6] W. S. Klug, M. R. Cummings. Concepts of Genetics, Prentice Hall, Upper Saddle River, 2003.

[7] I. Lin. Discovering Transcription Factor Binding Motif Sequences, Bioc218 Final Report, 2012.

[8] F. Zambelli, G. Pesole, G. Pavesi. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era, Brief Bioinform, 14(2), 225–37, doi:10.1093/bib/bbs016, 2013.

[9] D. P. Arce. Análisis in-silico de la expresión de genes sHsps en frutos de tomate *Solanum lycopersicum*, Trabajo Final presentado para optar al grado académico de Especialista en Bioinformática. Disponible en Biblioteca de la Facultad de Ciencias Agrarias UNR, 2016.

[10] M. Gismondi. Estudio in silico de la expresión génica relativa a factores protectores frente al daño por frío en duraznos. Trabajo Final presentado para optar al grado académico de Especialista en Bioinformática, 68 páginas. Disponible en Biblioteca de la Facultad de Ciencias Agrarias UNR, 2016.

[11] T. L. Bailey, M. Boden, F. Buske, M. Frith, C. E. Grant, L. Clementi, W. S. Noble. MEME SUITE: tools for motif discovery and searching, 2009.

[12] T. L. Bailey, C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings in the Second International Conference on Intelligent System for Molecular Biology, 28-36, http://www.sdsc.edu/~tbailey/papers/ismb94.pdf, 1994.

[13] S. Gupta, J. Stamatoyannopoulos, T. L. Bailey, W. S. Noble. Quantifying similarity between motifs, Genome biology 8:R24, doi:10.1186/gb-2007-8-2-r24, 2007.

[14] E. Portales-Casamar, S. Thongjuea, A. Kwon, D. Arenillas, X. Zhao, E. Valen, D. Yusuf, B. Lenhard, W. Wasserman, A-Sandelin. JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles. Nucleic Acids Research 38, issue SUPPL.1, D105–D110. doi:10.1093/nar/gkp950, 2009.

[15] A. Mathelier, O. Fornes, D. J. Arenillas, C. Chen, G. Denay, J. Lee, W. Shi, C. Shyr, G. Tan, R. Worsley-Hunt, A. W. Zhang, F. Parcy, B. Lenhard, A. Sandelin, W. W. Wasserman. JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles, Nucleic Acids Res 44 (D1): D110-D115, doi:10.1093/nar/gkv1176, 2015.

[16] J. M. Franco-Zorrilla, I. López-Vidriero, J. L. Carrasco, M. Godoy, P. Vera, R. Solano. DNA-binding specificities of plant transcription factors and their potential to define target genes, Proc Natl Acad Sci U S A 111(6), 2367–2372, doi:10.1073/pnas.1316278111, 2014.