

# **VCellID**

*Autores:*

*Alejandro Petit*

*Gisela de la Villa*

*Tutores:*

*Ariel Chernomoretz*

*Alan Bush*

*Alejandro Colman-Lerner*

*Fecha:*

*viernes, 23 de noviembre de 2012*

## Contenido

Plan de avance .....	3
Requerimientos del Proyecto .....	4
Requerimientos no Funcionales .....	6
Casos de Prueba .....	7
Arquitectura .....	9
Diseño de alto nivel.....	10
Casos de uso.....	10
Diseño de bajo nivel.....	10
Documentación de interfaces.....	11
ImageJ .....	11
VCellID.....	12
Image Setup .....	13
Load Images .....	14
Tree .....	15
Imagen .....	17
Menú.....	18
Segmentation.....	19
Progreso .....	21

## Plan de avance

1) 12/08/2011

Preparación del entorno de desarrollo e implementación de las siguientes ventanas, sin comportamiento:

- Main Window
- Cell ID Path
- Load Images
- About (automático con netbeans)
- Setup
  - Images Setup
  - Cell ID Dialog
- Segmentation Setup (Test)
- Run Cell ID Setup (Run)

2) 21/10/2011

- Modificación del entorno de desarrollo.
- Carga de imágenes con formato de nombre fijo.
- Creación del árbol de navegación.

3) 21/11/2011

- Ejecución y testeo con parámetros default.
- Creación de botones de navegación. Desplazamiento por canal, tiempo y posición. Anterior y siguiente.

4) 15/12/2011

- Ventana para definición de los parámetros de ejecución y agregado de opción para parámetros avanzados.
- Creación de barra de progreso para conocer el estado de la ejecución.

5) 16/04/2012

- Integración del manejo de imágenes con ImageJ.
- Sincronización de imágenes. Manejo de brillo, zoom y pan.
- Segmentación y generación de archivos para ejecución y testeo. Diferentes entornos.

6) 01/05/2012

- Revision 1

7) 23/05/2012:

- Flexibilización del nombre de las imágenes. Experimentos irregulares. Imágenes con y sin variable tiempo.
- Configuraciones particulares para contemplar distintas plataformas (S.O.)

8) 03/08/2012

- Revision 2

9) 08/09/2012

- Configuraciones finales. Nucleus. Fret.
- Ajustes de interfaz

10) 14/11/2012

- Entrega final

## Requerimientos del Proyecto

- Carga de experimentos
  - Flexibilizar nombres de imágenes (en load images)
  - Manejo de imágenes de 16 bits.
  - Permitir carga de experimentos no homogéneos
  - Poner memoria del último directorio usado
- Visualización de las imágenes
  - Por default, que la apertura de nuevas imágenes sean en una misma ventana (la activa)
  - Que se pueda abrir imágenes en ventanas nuevas para poder hacer comparaciones (por ejemplo, haciendo botón derecho -> "open in new window" o ctrl+click izquierdo)
  - Que se pueda determinar cuál de las ventanas abiertas es la activa
  - La imagen puede estar separada del árbol
  - Usar el visualizador de ImageJ con la idea de poder usar las otras funcionalidades de ImageJ sobre la imagen (imperativo: poder usar zoom, contraste y panning)
  - Esta posibilidad de edición tiene que ser sólo para visualizar
- Navegación
  - Botones de navegación por el árbol de imágenes (next para el mismo canal o cambio de canal para el mismo tiempo) : next position, next time, next channel.

- Cuando pasás por una categoría, las otras se tienen que mantener. Ej: estoy en pos.4, tiempo 7 y BF. Si paso por pos, voy a pos.5, tiempo 7 y BF. Si paso por tiempo, voy a pos.4, tiempo 8, BF. Si paso por filtro, voy a pos4, tiempo 7, FL (suponiendo que ese sea el filtro siguiente)
- Agregar checkbox "Synchro", tal que si está chequeado, los cambios de nextTime y nextPosition se hagan sobre todas las ventanas, no sólo la activa.
- Al estar chequeado "Synchro", el botón "next Channel" debería estar desactivado
- Si tengo una imagen ecualizada, que al pedirle nextTime y nextPosition me mantenga los parámetros de ecualización, así como el zoom y el pann.
- Que root haga un collapse all del árbol
- Si cambio de canal, recordar la modificación de brightness (HECHO)
- Experimentos no homogéneos
  - Contemplar escenarios donde una position tiene más tiempos que otra o un tiempo más canales
  - Si falta un canal o un tiempo, saltar al siguiente disponible. Ejemplo: si las posiciones 1, 2, 3 y 5 tienen 14 tiempos y la posición 4 tiene 10 y estoy navegando por posición el tiempo 13, al pedirle el siguiente a la posición 3 debería obtener el tiempo 13 de la posición 5. Con el código actual, estaría yendo al tiempo 13 de la posición 1.
  - Navegar por token y no por posición.
  - ALTERNATIVA: una vez recorrido el directorio y formado un árbol inicial, obtener el conjunto total de canales y el conjunto total de tiempos y completar los baches con imágenes que indiquen la inexistencia del dato requerido. Esto permite que el usuario note que no es una secuencia fluida, sino que hay un dato faltante.
- Test
  - Ver si efectivamente es necesario copiar el BF para el test o si se puede pasar dos veces el mismo path como parámetro a cellID (Alan)
  - Para correr test, hay que duplicar la imagen en una carpeta temporal y pasarla al cellID como canal de fluorescencia
  - Siempre sobre sólo una imagen (en particular un BF, que es lo que se usa para segmentar): habilitar el botón de test sólo si se está posicionado sobre una (run habilitado siempre)
  - Test NUNCA debería pisar a run! Con parámetro --output mandar a una carpeta tmp
- Ejecución
  - Run se puede hacer o bien para sólo una posición, o bien para todas las posiciones (por tiempo no se hace)
  - Si me paro en el root, corro todas las posiciones.
  - Si me paro en una posición o tiempo corro únicamente esa posición.

- Luego de ejecutar Test sobre una imagen, que se muestre la imagen resultante
- Luego de ejecutar Run sobre una rama, que se actualice el árbol, pero no se muestren las imágenes. (opc.)
- Permitir que se ejecute un experimento sin imágenes de fluorescencia (opc.)
- Armado de parámetros
  - bf as fl chequeado
    - se agrega el parámetro `treat_brightfield_as_fluorescence_also`
  - nucleus from channel chequeado
    - se crea el archivo `nuc_vcellid.txt` con el path de las imágenes y el nombre de archivo `3FP_Position1_time_001.tif` (por ejemplo, no?) 2 veces. Sería como el archivo de bf, pero con 3fp.
    - se agrega el parámetro `third_image nuclear_label`
  - splitted fret image chequeado y top seleccionado
    - se agregan los parámetros `fret bf_bottom_and_top` y `fret nuclear_top`
  - splitted fret image chequeado y bottom seleccionado
    - se agregan los parámetros `fret bf_bottom_and_top` y `fret nuclear_bottom`
  - Permitir agregar parámetros extra a la línea de comandos (opc.)
- Extras
  - Ver si lo que hacemos se puede invocar desde el lenguaje de scripting de ImageJ
  - Poner expresiones regulares para los patrones de los nombres de imágenes
  - Agregar un botón para clonar la imagen activa en otra ventana y poder comparar imágenes

## Requerimientos no Funcionales

- Sistema
  - Debe ser compatible con Windows y con Unix
  - Setear locale EN (el procesador cell no entiende locale ES)
  - Integrar con ImageJ. Ver de contener todos los menús en una ventana. Básicamente, la aplicación original sin mostrar la imagen, sino que pasársela a ImageJ para que la muestre.
- Estética
  - Los botones tienen que estar organizados 3x2 (es decir 3 filas y 2 columnas)
  - Tiene que estar por categorías en las filas (pase por tiempo, pase por posición y pase por filtro)
  - Hacer los botones todos del mismo tamaño y con espacio entre cada botón

- Remover el actual panel de visualización de imágenes, de manera que el árbol de navegación quede lo más angosto posible y haya lugar para desplegar cómodamente varias imágenes
- Hacer un progress bar para ver el avance del procesamiento de las imágenes. Alternativa: ruedita, para ver que sigue procesando.
- Poner botón test y run juntos
- El "open in new window" hacerlo en el menú
- En segmentation, mostrar un textbox con parámetros adicionales para la línea de comando que pueda ser modificada, con un checkbox que permita habilitarlo o no
- Si time o position están vacíos, completarlos con grises
- Si no existe una imagen, cambiarle el ícono de archivo o poner letra en gris, para que pase más desapercibido. Se puede agregar un atributo al nodo.
- Lanzar Errores
  - Si el canal especificado para nucleus from channel es erróneo

## Casos de Prueba

Los casos de prueba que se llevaron a cabo, fueron sumamente variados. Esto se debió a la dificultad de implementar una interfaz gráfica en múltiples plataformas y sin un entorno que facilite su inmediata visualización. La mayoría de las pruebas de usabilidad y formato visual, se llevaron a cabo sin una herramienta de testeo.

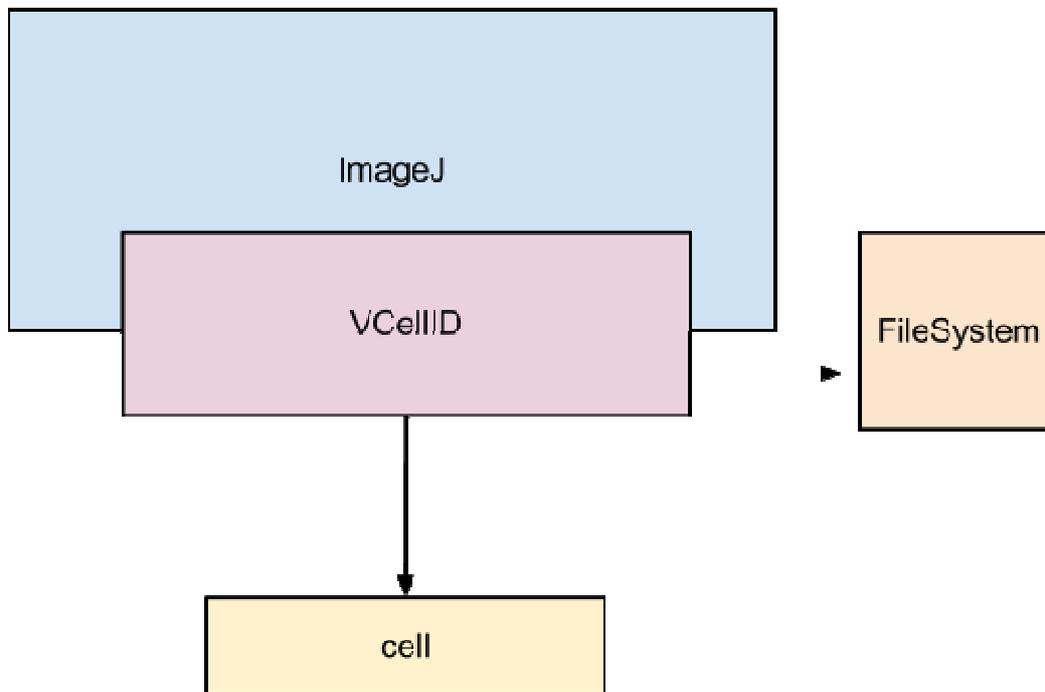
Caso de Testo	Escenario	Pruebas	Resultados Esperados
1	No se especifico el ejecutable del cellid	Selección de la opción path en la ventana del cellid	Identificación del sistema operativo y selección del directorio por default de acuerdo al mismo.
2	Desconocimiento de formato del cellid	Selección de archivo o directorio dentro de la opción CellldPath	No se permite seleccionar un directorio o archivo que no sea ejecutable para la plataforma correspondiente
3	Reutilización de parámetros de testeo o ejecución	Selección de opción "OK" o "Apply" en las ventanas de configuración	El botón "OK", mantiene los parámetros y lanza la ejecución. El botón "Apply", guarda el contexto sin lanzar la ejecución. "Cancel" elimina los cambios.

4	Los parámetros de ejecución están incompletos	Selección de imágenes de corrección, sin identificar el nombre de las mismas.	Mensaje de error, informando el dato requerido.
5	Carga de imágenes con formato de nombre inválido	Definición de nombre de imagen y búsqueda en directorio	Se genera el árbol con un único nodo, el raíz, ya que no se encuentran imágenes con el patrón especificado
6	Carga de experimentos incompletos	Definición de nombre de imagen y selección de directorio	A aquellos tiempos y posiciones que no tengan una imagen asociada se les asigna una imagen vacía, para completar el experimento
7	Selección de imagen del árbol	Click derecho o izquierdo sobre las hojas del árbol	Apertura de la imagen seleccionada a través del ImageJ. Si la imagen no existe, se abre una por defecto.
8	Selección de opciones sobre los elementos del árbol	Click derecho o izquierdo sobre cualquier nodo del árbol	Dependiendo el tipo de nodo, las opciones que se presentan. Ejecución normal, ejecución BF, testeo, apertura de imagen en nueva pestaña.
9	Navegabilidad del árbol	Utilización de los botones "Next" y "Previous"	Selección de la imagen correspondiente, de acuerdo a la acción indicada. De las tres coordenadas (Posición, tiempo, canal), únicamente se modifica la indicada.
10	Ajustes sobre imágenes	Utilización de ImageJ, para cambiar el brillo, zoom y pann de las imágenes.	Modificación de la imagen y aplicación de la corrección en la navegación de las mismas. En el caso del canal, cada uno tiene su brillo.
11	Sincronización de imágenes	Al tener varias imágenes abiertas, navegar el árbol, con la opción "Sync"	Todas las imágenes sufren el mismo desplazamiento, en la

		activada	coordenada indicada.
12	Ejecución y testeo	Selección de opción "Run ..." o "Test ..." sobre algún nodo del árbol	Se abre la pantalla de Segmentación para configurar los parámetros de ejecución
13	Parámetros avanzados	Selección de la opción parámetros avanzados en la ventana de Segmentación	Los parámetros agregados se concatenan tal cual a la llamada al cellid
14	Ejecución para test.	Ejecutar la aplicación en modo test	Se generan los archivos y las imágenes correspondientes en un directorio "Test", para no modificar la integridad de la ejecución normal.
15	Barra de progreso de ejecución	Iniciar la ejecución del cellid.	La barra de progreso muestra cuando se finaliza la corrida de cada posición.

## Arquitectura

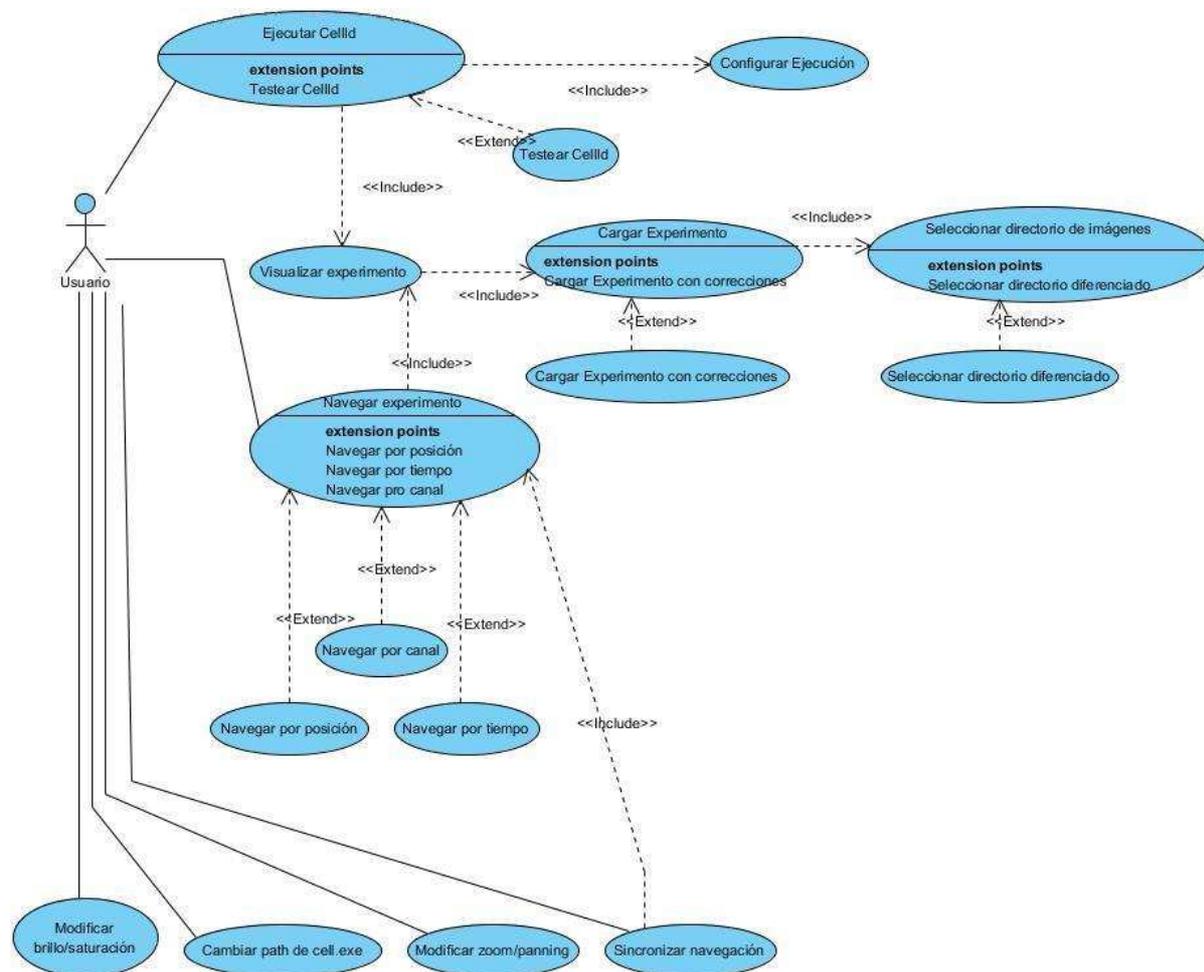
La aplicación original de VCellID fue desarrollada como plugin del programa ImageJ.



Para hacer uso de VCellID, el usuario primero debe abrir la aplicación ImageJ. Luego de abrir el plugin VCellID, se interactúa con el filesystem para cargar los experimentos (el usuario debe seleccionar el directorio donde éste se encuentra). Una vez cargado el experimento, el usuario puede querer analizar las imágenes y buscar las células en ellas. Para ello, el sistema interactúa con el programa cell, invocandolo con los parámetros especificados por el usuario. Finalmente, el sistema interactúa nuevamente con el filesystem para registrar la salida de la ejecución de cell.

## Diseño de alto nivel

### Casos de uso

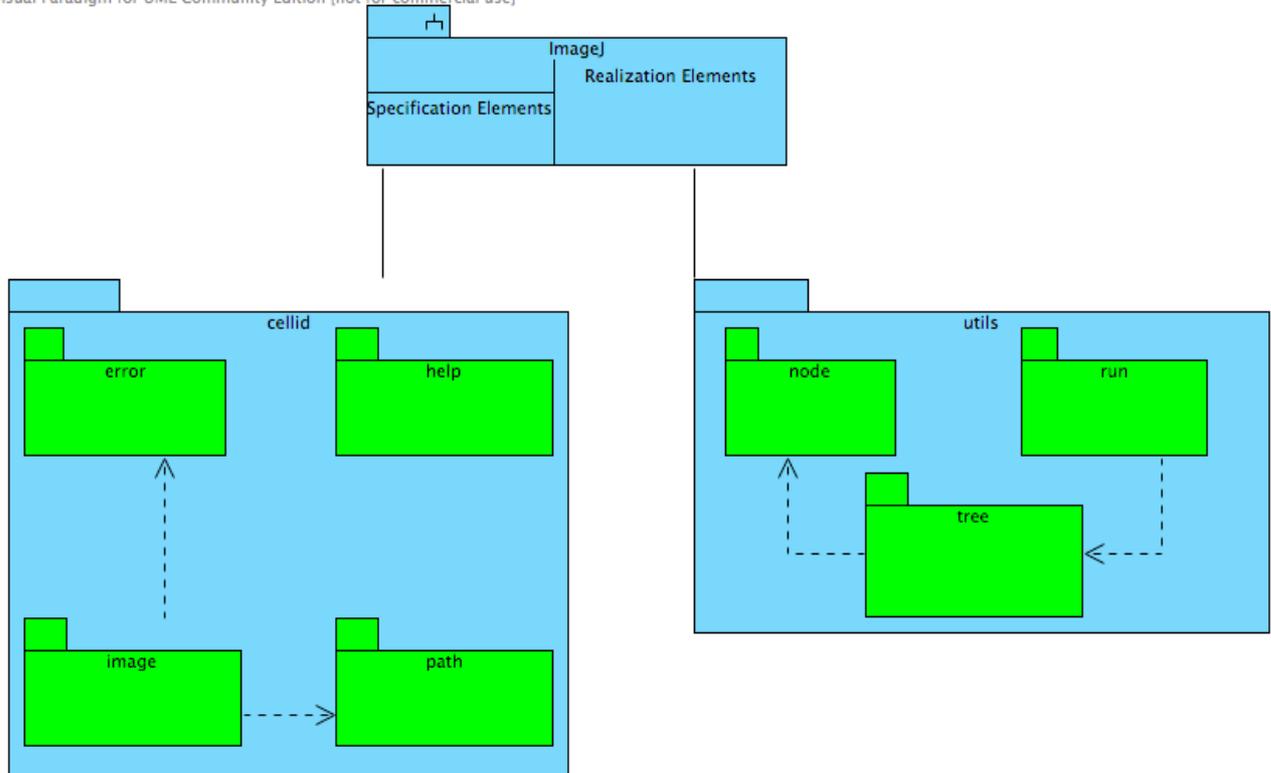


## Diseño de bajo nivel

El siguiente diagrama muestra la división de los paquetes que conforman la aplicación. Podemos observar una primera división entre interfaz, paquete cellid y lógica de la aplicación, paquete utils. Dentro del primero se puede observar la división por cada ventana de la

aplicación y en el segundo, por la función que desempeña cada uno. Desde la confección del árbol, a la generación de los archivos y la ejecución de cellid.

Visual Paradigm for UML Community Edition [not for commercial use]



Entrando más en detalle, podemos observar todas las clases de las que dispone la aplicación y cómo se relacionan entre sí. Es interesante notar también, la estructura seleccionada para el manejo de imágenes. Dada la fuerte relación entre la definición del usuario con el nombre de las imágenes, se decidió crear una clase que maneje esta información y que sea la encargada de especificar el formato de la misma. La estructura de Imágenes, PositionImage, Time Image, resulta esencial para el armado del árbol, ya que son independientes de formato de las imágenes en el sentido que los tokens identificadores los maneja otra clase aparte, con la cual interactúan profundamente.

En lo que concierne a la construcción de árbol, nuevamente vemos una profunda interacción entre los componentes para sincronizar la visualización del árbol, con su comportamiento.

Finalmente para la ejecución, es necesario mantener el contexto de ejecución no sólo para generar los archivos de entrada correspondiente y los parámetros de configuración, sino para una vez finalizada la ejecución, conocer las acciones a tomar.

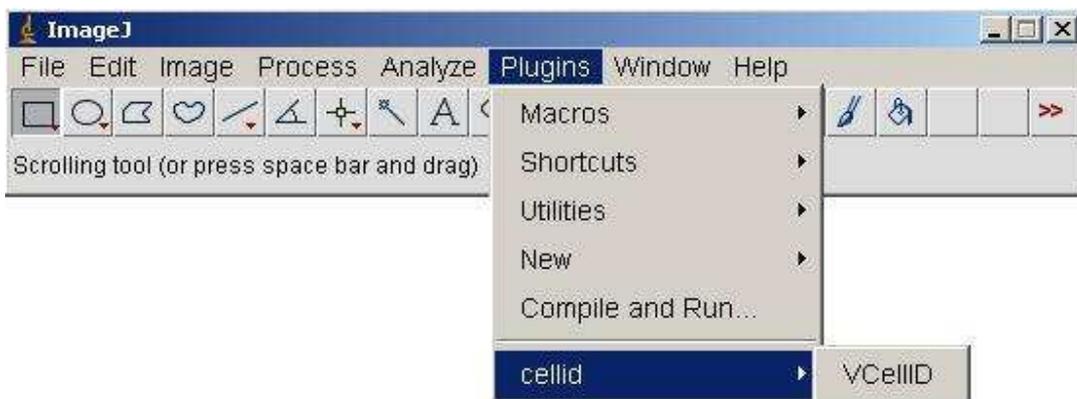
## Documentación de interfaces

### ImageJ

Esta ventana es lo primero con lo que el usuario se encuentra al querer ejecutar VCellID. Como está desarrollado como plugin de ImageJ, es esto lo que deben abrir primero.



Se accede a VCellID yendo a Plugins>cellid>VCellID.



## VCellID

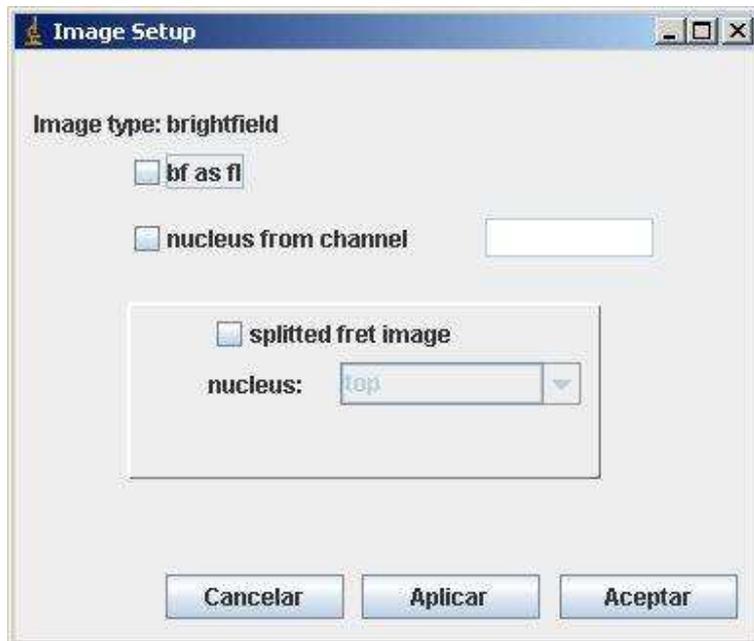


Como ImageJ está diseñado con secciones desacopladas, al seleccionar el plugin VCellID se abre una nueva ventana con este menú.

En ella se ven cuatro opciones:

- CellID Path: permite determinar la ubicación del ejecutable cell, a quien se le pasarán luego los parámetros para analizar las imágenes y detectar las células.
- Images Setup: abre una sección donde se puede especificar ciertas propiedades del experimento (a desarrollar en su sección)
- Load Images: despliega la ventana donde se eligen las rutas para las imágenes a analizar, así como sus imágenes de fluorescencia y las imágenes de corrección.
- Help: abre una ventana con instrucciones para obtener ayuda.

## Image Setup



En esta ventana se pueden especificar las siguientes propiedades del experimento:

- **bf as fl:** indica que las imágenes brightfield (es decir, aquellas a analizar) deben ser tomadas como su propia imagen de fluorescencia también. Agrega el parámetro bf-as-fl.
- **nucleus from channel:** se usa para encontrar núcleos etiquetados fluorescentemente con Cell-ID. A continuación se debe insertar el canal del cual se toman los núcleos. De no existir el canal, se lanza un error. Agrega el parámetro `third_image nuclear_label` y se crea el archivo `nuc_vcellid.txt`, que adentro tiene repetido el nombre de la tercer imagen tantas veces como canales haya.

Este archivo tiene formato de nombre de imagen de fluorescencia, sólo que en lugar del identificador de canal tiene un 3 (por ejemplo: si tuviéramos imágenes nombradas `<canal>_Position<posición>_time_<tiempo>.tif`, y siendo FP el denotador de imágenes de fluorescencia, el nombre de la imagen sería `3FP_Position<posición>_time_<tiempo>.tif`). Esta imagen no es provista por el usuario, sino que es una copia de la imagen del canal especificado para núcleo.

- **splitted fret image:** sirve para analizar un experimento FRET con imagen dividida. Se agregan el parámetro `fret bf_bottom_and_top` y, según esté elegido top o bottom, el parámetro `fret nuclear_top` o `fret nuclear_bottom`.

Los comportamientos de los botones son:

- **Cancelar:** cierra la ventana sin aplicar cambios.
- **Aplicar:** los cambios se guardan pero la ventana permanece abierta.
- **Aceptar:** se aplican los cambios y se cierra la ventana.

Recordemos que son ventanas flotantes, con lo cual uno podría querer correr varios experimentos modificando estas propiedades, para lo cual no necesitaría abrir y cerrar esta ventana todo el tiempo.

## Load Images

**Load Images Patterns**

Position token: Position  example: fluorescence file

Time token: time\_  ?FP\_Position(d\*)\_time\_(d\*).tif

Character separator:  \_

	token	path	
bright field	<input type="text"/> BF	<input type="text"/>	<input type="button" value="Open..."/>
fluorescent	<input type="text"/> FP	<input type="text"/>	<input type="button" value="Open..."/>
<input type="checkbox"/> uneven illumination correction image (basename)	<input type="text"/>	<input type="text"/>	<input type="button" value="Open..."/>
<input type="checkbox"/> camera background correction image (basename)	<input type="text"/>	<input type="text"/>	<input type="button" value="Open..."/>

force same path

group images by:  file name pattern

Esta ventana se utiliza para la carga de imágenes. Los parámetros son:

- Position token: indica cuál será el identificador de la posición dentro del nombre de imagen.
- Time token (opc): indica el identificador del tiempo en el nombre de imagen.
- token: indica los identificadores de imágenes brightfield e imágenes de fluorescencia.
- path: indica el directorio de donde se debe cargar el experimento. Por default se cargan todas las imágenes desde el mismo directorio. Si en cambio deshabilitamos la opción "force same path", podemos diferenciar el directorio de imágenes brightfield del directorio de imágenes de fluorescencia y de los directorios de los archivos de corrección.
- En el drop-down al lado de "group images by" se selecciona cómo se debe mapear Cell-ID las imágenes BF a imágenes de fluorescencia. Esto no está implementado, oficia de placeholder para continuar el desarrollo.

- camera background correction image y uneven illumination correction image(opc): sirven para aplicar corrección a las imágenes cuando la iluminación del microscopio no es uniforme. en los campos de texto se debe especificar el nombre de archivo en el cual constan las imágenes de corrección (una por canal). Usualmente se llamarán flat.txt para uneven illumination y dark .txt para camera background.

Los botones cumplen las siguientes funciones:

- OK: cierra la ventana y carga un nuevo árbol con el experimento
- Cancel: Cierra la ventana sin cargar el árbol

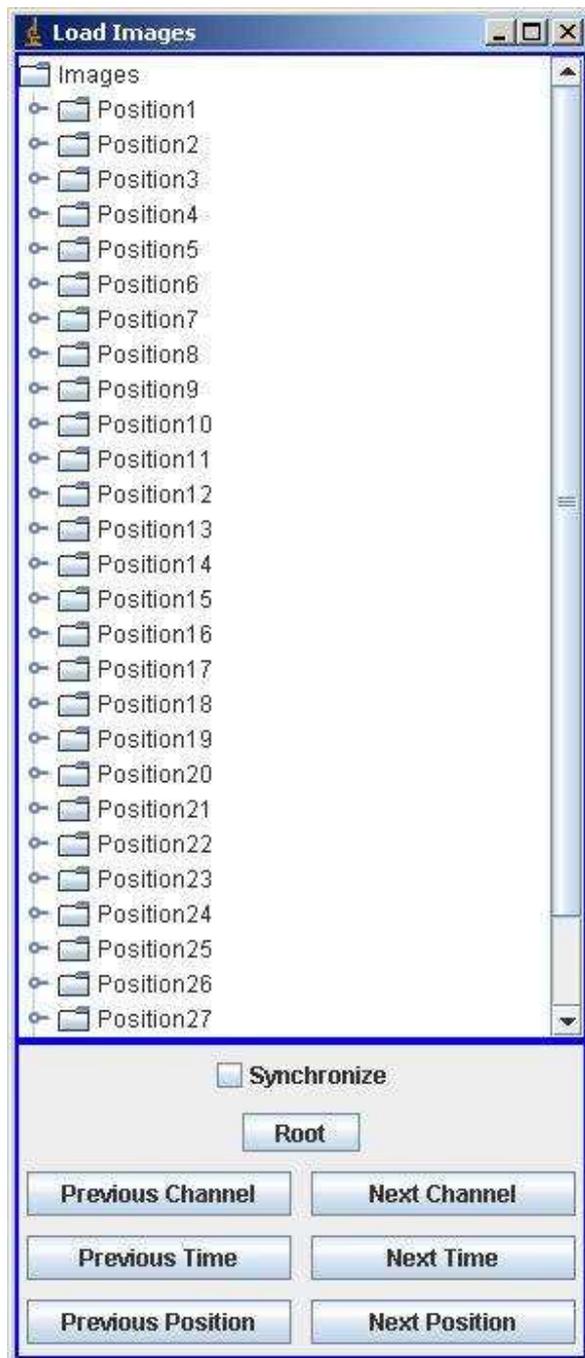
## Tree

En este árbol se muestra el experimento en su totalidad. Esta es la sección que más diferencias presenta respecto de su versión original.

En primer lugar, el árbol está desacoplado. Como no pertenece a una ventana única de aplicación, se puede tener varios experimentos abiertos a la vez. A su vez, la imagen seleccionada también se encuentra desacoplada, con lo cual podemos abrir varias imágenes simultáneamente. Esto presenta la ventaja de poder comparar imágenes, cosa que antes no se podía por la imposibilidad de abrir más de una imagen a la vez.

Luego, tenemos el panel de navegación. Esta es otra mejora respecto de la versión original. Mientras que anteriormente había que seleccionar las imágenes con el ratón (lo que aún se puede hacer), ahora se presentan las siguientes posibilidades:

- Navegación por canal: Botones “Previous Channel” y “Next Channel”. Dentro de un mismo tiempo y posición, se cicla sobre los canales. Si al llegar al último se pide el siguiente, se va al primero.
- Navegación por tiempo: Botones “Previous Time” y “Next Time”. Dentro de una misma posición, se cicla sobre los tiempos, manteniendo el canal seleccionado. Si al llegar al último tiempo se pide el siguiente, se va al primero.
- Navegación por posición: Botones “Previous Position” y “Next Position”. Se cicla sobre las posiciones, manteniendo tiempo y canal seleccionados. Si al llegar a la última posición se pide la siguiente, se va a la primera.
- Root: El botón “Root” colapsa el árbol y hace que sea la raíz lo que esté seleccionado. La raíz es un nodo ficticio.



Con la posibilidad de tener varias imágenes abiertas, surge una nueva posibilidad: sincronizarlas para su navegación. Esto es lo que hace el checkbox "Synchronize". Habiendo varias imágenes abiertas, si chequeo "Synchronize" puedo navegar en todas simultáneamente.

Supongamos que quisiéramos ver un área más grande del grupo celular y analizar en todo el conjunto qué sucede tiempo a tiempo. Uno querría abrir imágenes para el mismo tiempo y

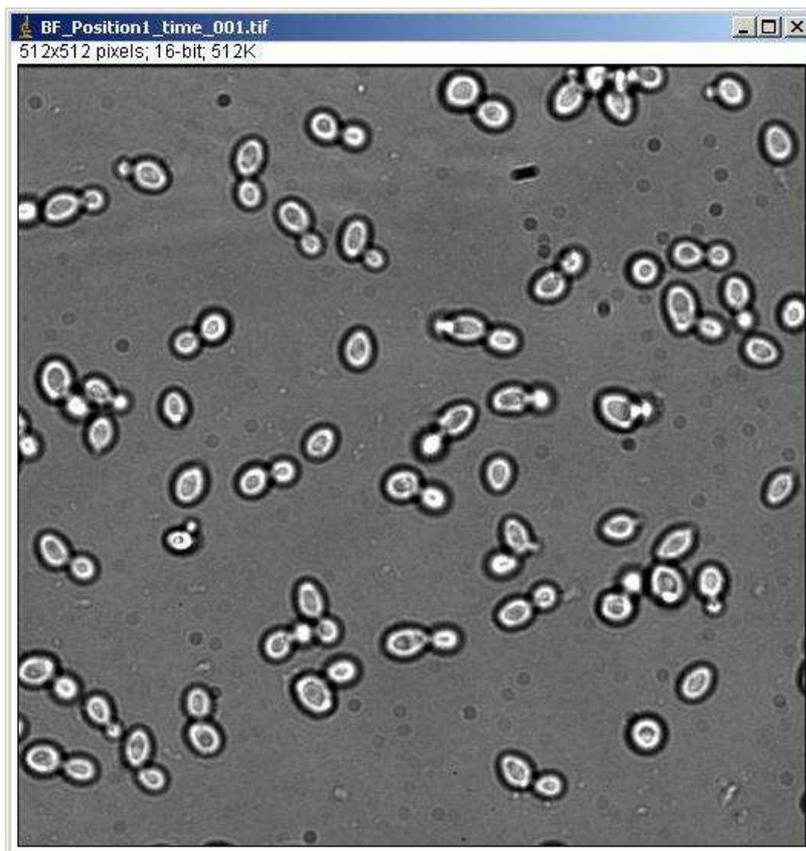
canal para posiciones contiguas y sincronizarlas para navegar por tiempo en todas a la vez, ver el avance en simultáneo.

En caso de estar seleccionado "Synchronize", se puede navegar secuencialmente por tiempo o por posición. En este caso, la navegación secuencial por canal está deshabilitada, dado que es una funcionalidad que no tiene sentido en sincronía.

## Imagen

Para visualizar las imágenes, se usa la ventana que provee imageJ. Esta ventana está desacoplada del resto, lo que permite abrir varias a la vez y determinar cuál es la activa con sólo seleccionarla.

Utilizar la ventana de ImageJ trae aún más ventajas sobre el visualizador que el antiguo VCellID tenía. Como pertenece a ImageJ, se pueden utilizar todas las funcionalidades del programa para modificar la imagen. Esto hace que sin necesidad de implementar nada nuevo tengamos funcionalidades como el ajuste del brillo y la saturación, así como el zoom y el pann. Además, ImageJ contempla ya el manejo de imágenes de 16-bit, que es algo que el VCellID original no tenía. La salida del programa cell es en 16-bit y sólo se podía visualizar con una aplicación externa.

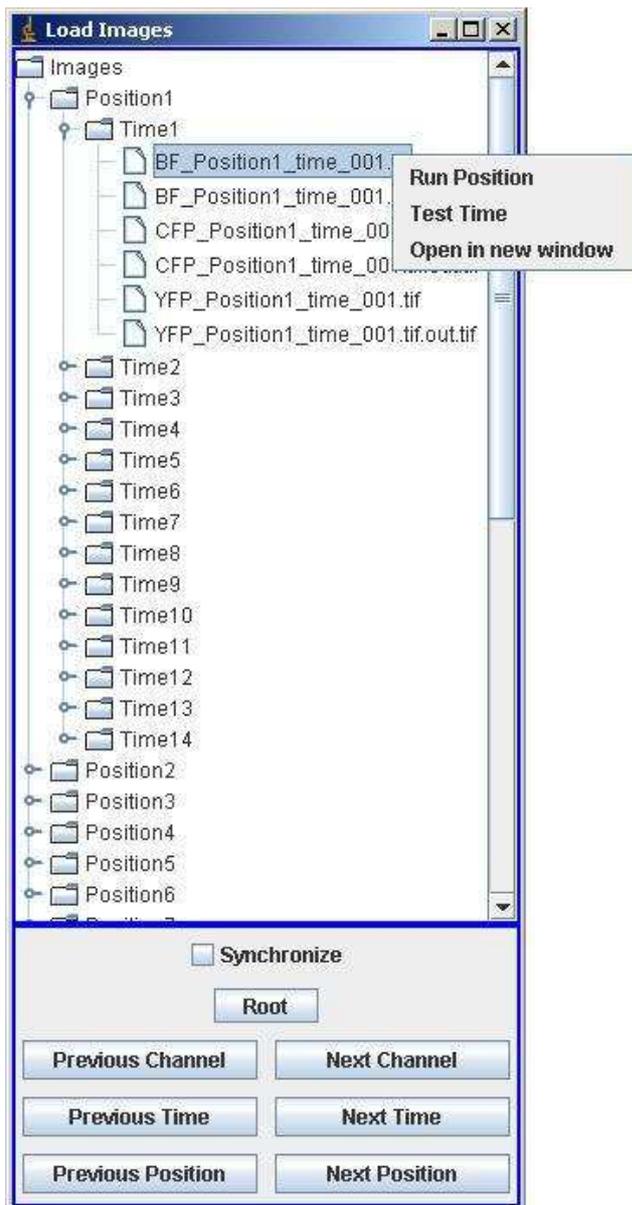


La ventana de imagen en sí recuerda el pann y el zoom, con lo cual al abrir una imagen en una misma ventana, estos valores se mantienen. Sin embargo, esto no sucede para el brillo y saturación. Mientras que las primeras propiedades son propiedades de la ventana, las últimas

son propiedades de la imagen. Es por ello que la memoria de zoom y pann en la navegación se obtuvo con el simple hecho de utilizar la ventana de imágenes de ImageJ, mientras que para brillo y saturación hubo que manejar la memoria especialmente.

## Menú

El menú se despliega haciendo click derecho sobre un elemento del árbol. La imagen es a modo ilustrativo de cómo se ve el menú. Éste varía según el elemento del árbol sobre el que se haga click.

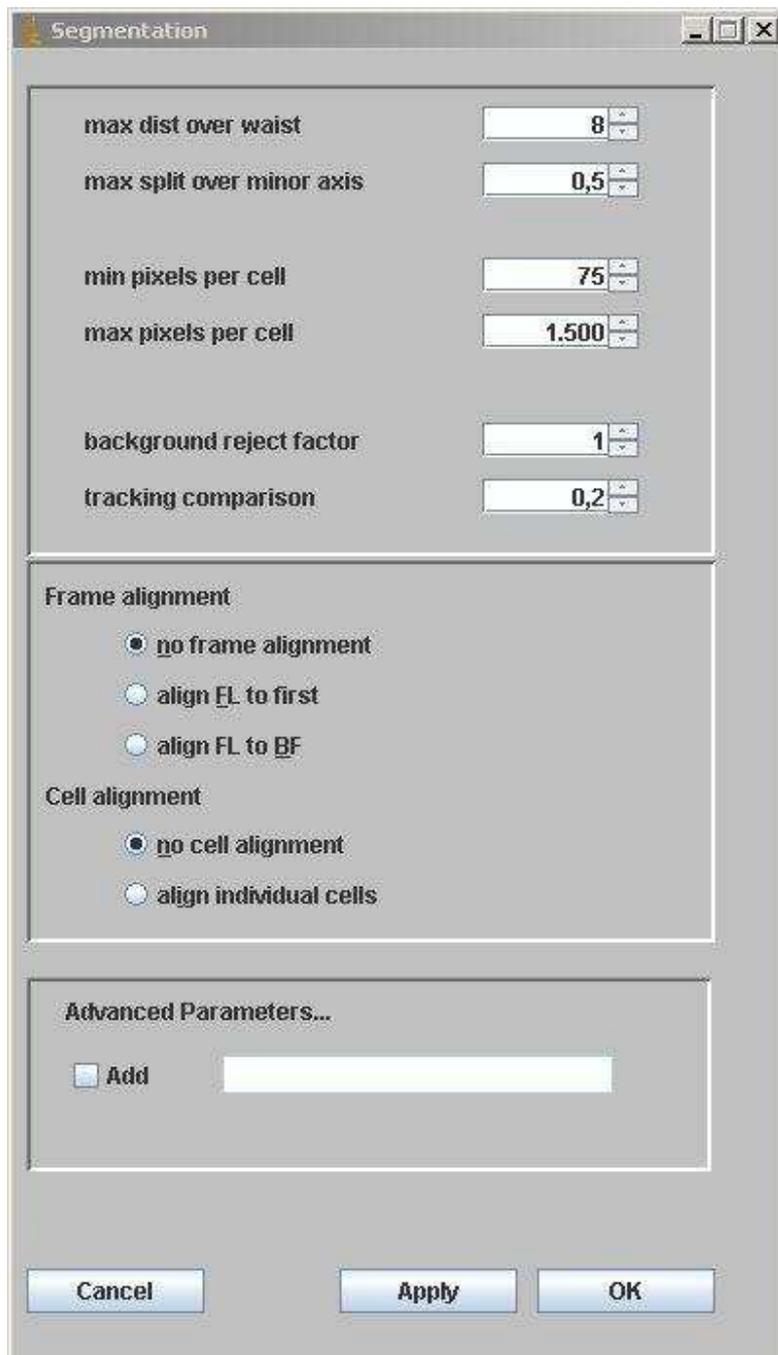


A continuación, detallamos los ítems por tipo de elemento del árbol:

- Images (Root)
  - Run: Corre todo el experimento normalmente.
  - Run BF: Corre todos los tiempos de todas las posiciones solo con BFs. Es equivalente a hacer Test para todas las posiciones.
- Position
  - Run: Corre todos los tiempos de la posición
  - Run BF: Corre todos los tiempos de esa Posición solo con BFs.
- Time
  - Run Position: Corre la posición a la que pertenece el tiempo, para todos los tiempos de esa posición.
  - Test Time: Testea este tiempo.
- BF
  - Run Position: Corre la posición
  - Test Time: Testea este tiempo.
  - Open in new window: Abre la imagen en una ventana nueva.
- FP y .out
  - Open in new window: Abre la imagen en una ventana nueva.

## Segmentation

Esta es la ventana que se abre cada vez que le damos Run o Test a un nodo. En ella hay diferentes parámetros que impactarán fuertemente en el resultado del reconocimiento celular.



Los parámetros son:

- “max dist over waist” y “max split over minor axis”: Estos parámetros están diseñados para determinar si una célula que se está reproduciendo debe ser dividida en dos: madre e hija.
- “min pixels per cell” y “max pixels per cell”: Son límites inferior y superior para el tamaño de la célula, en número de píxeles. Este parámetro es útil para limitar el número de células espúreas.

- “background reject factor”: El algoritmo de segmentación de Cell-ID calcula inicialmente cuál debería ser la intensidad de los píxeles del borde de las células como la media de las intensidades de la imagen BF y le sustrae un número definido de desvíos estándar. Entonces, inicialmente considera que los píxeles que tengan intensidad por debajo de este número pertenecen a bordes. “background reject factor” es el número de desvíos estándar que se sustrae. Sus valores usuales son entre 0,4 y 1,2. Para imágenes que estén levemente fuera de foco, conviene tomar un número mayor, pues reducirá el número de células espúreas. Sin embargo, si los bordes son muy delgados convendrá que el número sea menor, a pesar de que aumente el número de resultados espúreos.
- “tracking comparison”: Cell-ID intenta rastrear células a lo largo del tiempo. El valor de este parámetro es la superposición fraccional mínima entre dos células en tiempos consecutivos para que sean consideradas la misma célula. El valor default es 0,2.
- “Frame alignment”: Cell-ID puede mover una imagen en el plano xy para alinearla con otra imagen de referencia. Si está seleccionado “align FL to BF”, la imagen BF es usada como referencia. Si está seleccionado “align FL to first”, la primer imagen de fluorescencia es usada como referencia.
- “Cell alignment”: si “align individual cells” es seleccionado, Cell ID va a mover el contorno de cada célula encontrada en la imagen BF para que coincida con las células en las imágenes de fluorescencia. Esta función es útil cuando las células se mueven ligeramente entre cada adquisición de imágenes. Esta opción aumenta el tiempo de ejecución de Cell-ID significativamente y no debería ser utilizada si las células están fijas.
- Advanced parameters: Esta opción fue incluida como nuevo requerimiento en esta nueva versión de VCellID. Un usuario avanzado podría querer utilizar parámetros de Cell-ID no configurables vía la aplicación gráfica. Es por esto que se le facilita este campo. Si se chequea la opción “Add”, los parámetros que estén incluidos en el campo de texto serán agregados a la invocación de Cell-ID. Si esta opción está deseleccionada, el texto sigue almacenado y los parámetros no se agregan. Esto permite al usuario hacer diferentes pruebas rápidas con y sin parámetros extra sin la necesidad de agregarlos cada vez.

Las acciones de los botones son:

- Cancel: Cierra la ventana sin almacenar los cambios de los parámetros.
- Apply: Almacena los cambios sin cerrar la ventana.
- OK: Almacena los cambios y ejecuta el run o test, según de dónde provenga el pedido.

## Progreso

Para mostrar el progreso de un experimento, se implementó una barra de porcentaje con log. Éstos se actualizan al terminar de procesar una posición.

