Getting started with ReadXplorer

This guide is intended for beginners using ReadXplorer for the first time. It will give a short introduction about the main features of the software and how they can be used. Please be aware that this guide is not intended to replace the complete manual.

Content

1. What is ReadXplorer?

ReadXplorer is a viewer and automatic analysis platform for Next-Generation Sequencing data. It can be used to display a reference genome sequence with genomic features (annotations) alongside with different tracks that represent read data that was mapped against the reference. In contrast to most other viewers, ReadXplorer also offers a wide range of analysis functions that can be applied to the visualized data.

2. When to use ReadXplorer?

The Software can be used with every data-set created by mapping read data against a reference sequence. This can be for example a resequencing experiment or an RNA sequencing run.

3. What input data is needed?

The software has to be provided with both a reference sequence and the corresponding read mapping data.

The reference sequence can be provided in Genbank, EMBL or Fasta format. If the Fasta format is used corresponding genomic features (annotations) can be added in GTF/GFF2 or GFF3 format. It is advisable to import genomic features because most analysis functions only work properly when they are present. Nevertheless, ReadXplorer will also accept reference sequences without any annotation.

The mapped reads must be provided in sam/bam format. These files must have been created by mapping the read data against the exact same reference as the one imported into ReadXplorer. This is extremely important: during import the mapping files will be assigned to a reference. If the reference noted down in the sam/bam file does not match the reference sequence it is assigned to no reads will be visible! You are free to use the mapping tool you prefer (e.g. Bowtie, bwa, Tophat) as long as the output is in standard conform sam/bam format[1]. Mapping data imported is called a *track* within the software. ReadXplorer is primarily designed for Illumina short-read data (single- or paired-end) but is not limited to it. As long as the read data can be mapped against a reference and the output is in sam/bam format ReadXplorer can handle the data. Of course not all analysis functions offered by ReadXplorer are sensible to use with every kind of input data. However, the software is not designed to patronize its user unless it is inevitable and hence all analysis functions can be used with any read mapping data-set.

[1] https://samtools.github.io/hts-specs/SAMv1.pdf

4. Which installer should I choose?

The ReadXplorer homepage provides various installers suitable for different setup scenarios:

Zip-x86/64-AllSystems

This is actually no installer but simply a zip file containing the ReadXplorer software which can easily be used on all operating systems. It can be unzipped and ReadXplorer can be started by executing one of the binaries in the "bin" directory. There are different binaries for Windows and Linux in this directory. You must install an appropriate version of Java on your own before you can start the program.

Exe-Windows-i386-incl Java

This is a standalone installer for Windows x86. It requires at least Windows 7. The installer comes with its own Java version and hence no Java needs to be pre-installed on your system. Java will only be installed for ReadXplorer. This means that if you already have a Java installation it will not be altered. This installer comes with the 32bit version of Java. It will run on 32bit Windows systems but you can only use a limited amount of RAM (heap-size):

> The maximum theoretical heap limit for the 32-bit JVM is 4G. Due to various additional constraints such as available swap, kernel address space usage, memory fragmentation, and VM overhead, in practice the limit can be much lower. On most modern 32-bit Windows systems the maximum heap size will range from 1.4G to 1.6G. On 32-bit Solaris kernels the address space is limited to 2G. On 64-bit operating systems running the 32-bit VM, the max heap size can be higher, approaching 4G on many Solaris systems.[2]

Exe-Windows-x64-incl Java

This is a standalone installer for Windows x64. It requires at least Windows 7. The installer comes with its own Java version and hence no Java needs to be pre-installed on your system. Java will only be installed for ReadXplorer. This means that if you already have a Java installation it will not be altered. This installer comes with the 64bit version of Java and hence it can only be installed on a 64bit Windows. **This is the recommended way of installing ReadXplorer on a Windows computer.**

Linux-i386-incl Java

This is a standalone installer for Linux. It should work on all major Linux distribution although there are so many different distributions available that we cannot guarantee that. We test the Linux installers on Fedora 23. The installer comes with its own Java version and hence no Java needs to be pre-installed on your system. Java will only be installed for ReadXplorer. This means that if you already have a Java installation it will not be altered. This installer comes with the 32bit version of Java. It will run on 32bit Linux systems but you can only use a limited amount of RAM (heap-size):

> The maximum theoretical heap limit for the 32-bit JVM is 4G. Due to various additional constraints such as available swap, kernel address space usage, memory fragmentation, and VM overhead, in practice the limit can be much lower. On most modern 32-bit Windows systems the maximum heap size will range from 1.4G to 1.6G. On 32-bit Solaris kernels the address space is limited to 2G. On 64-bit operating

[2] http://www.oracle.com/technetwork/java/hotspotfaq-138619.html#gc_heap_32bit

systems running the 32-bit VM, the max heap size can be higher, approaching 4G on many Solaris systems.[2]

The installer is compressed. Once downloaded open a terminal and navigate to the download folder. Decompress the installer:

- `gunzip readxplorer_2.2-linux-i386.sh.gz`

Afterwards you can launch the extracted sh file:

- `sh ./readxplorer_2.2-linux-i386.sh`

Linux-x64-incl Java

This is a standalone installer for Linux. It should work on all major Linux distribution although there are so many different distributions available that we cannot guarantee that. We test the Linux installers on Fedora 23. The installer comes with its own Java version and hence no Java needs to be pre-installed on your system. Java will only be installed for ReadXplorer. This means that if you already have a Java installation it will not be altered. This installer comes with the 64bit version of Java and will only run on 64bit Linux systems. **This is the recommended way of installing ReadXplorer on a Linux computer.**

The installer is compressed. Once downloaded open a terminal and navigate to the download folder. Decompress the installer:

- `gunzip readxplorer_2.2-linux-x64.sh.gz`

Afterwards you can launch the extracted sh file:

- `sh ./readxplorer_2.2-linux-x64.sh`

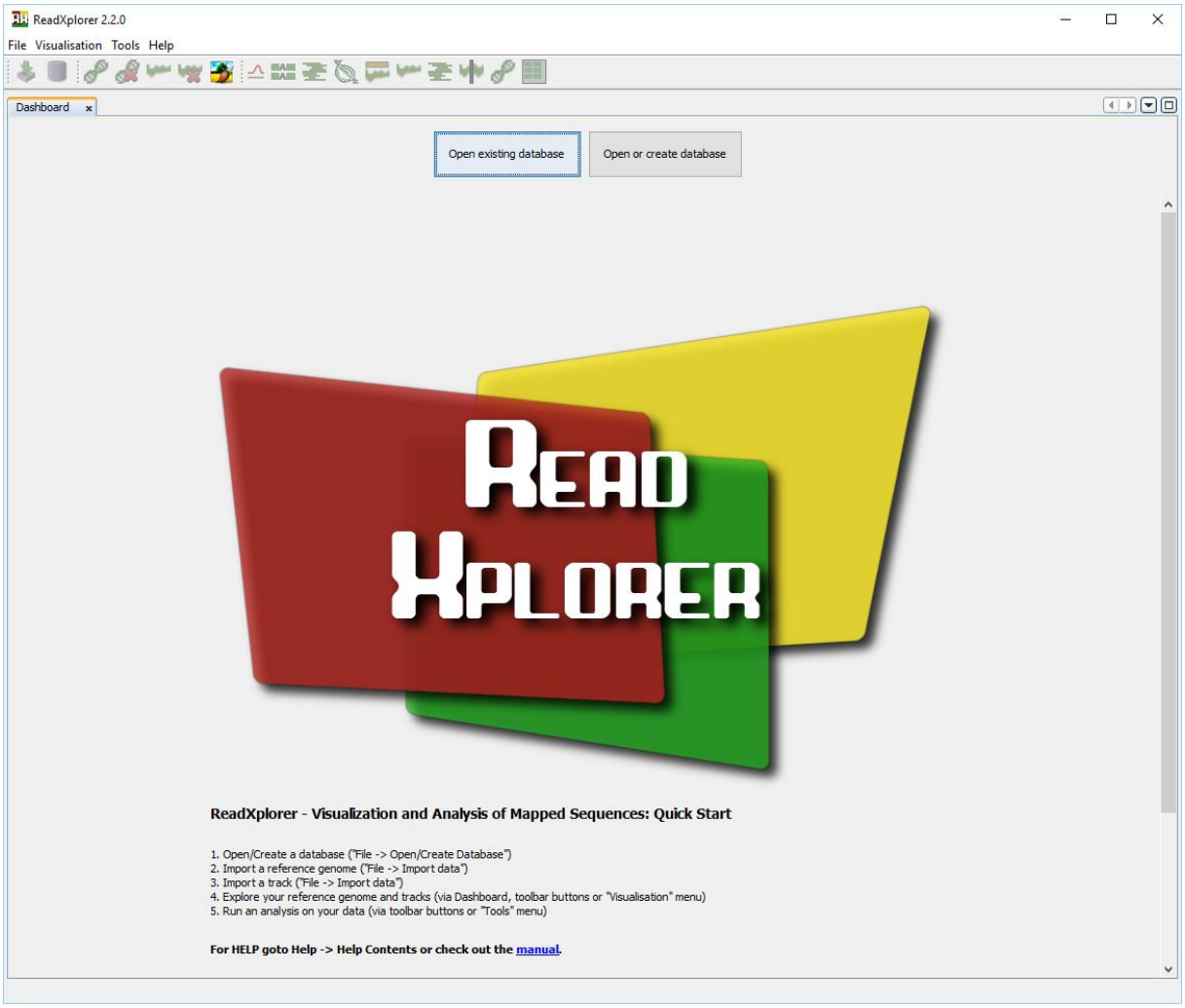
DMG-Mac-incl Java

This is a disk image file for OS X. Like most Mac programs it does not have a real installer. The DMG is simply opened and then the application is installed by "drag and dropping" it to the "Application" folder. Java is included in the ReadXplorer application and hence no Java needs to be pre-installed, an existing Java Version will not be altered. Additionally, GNU R (needed for the differential expression analysis) is included in this installer. A GNU R instance is automatically launched when you start ReadXplorer and automatically stopped when you close ReadXplorer. We utilize RServe listening on port 6311 to make GNU R accessible from within Java. Please make sure that no other RServe instance is running when you launch ReadXplorer as this might interfere with the bundled GNU R. If you never heard of RServe, don't worry, you will most probably be safe to launch ReadXplorer without taking care of anything. **This is the recommended way of installing ReadXplorer on OS X computer.**

5. The first run

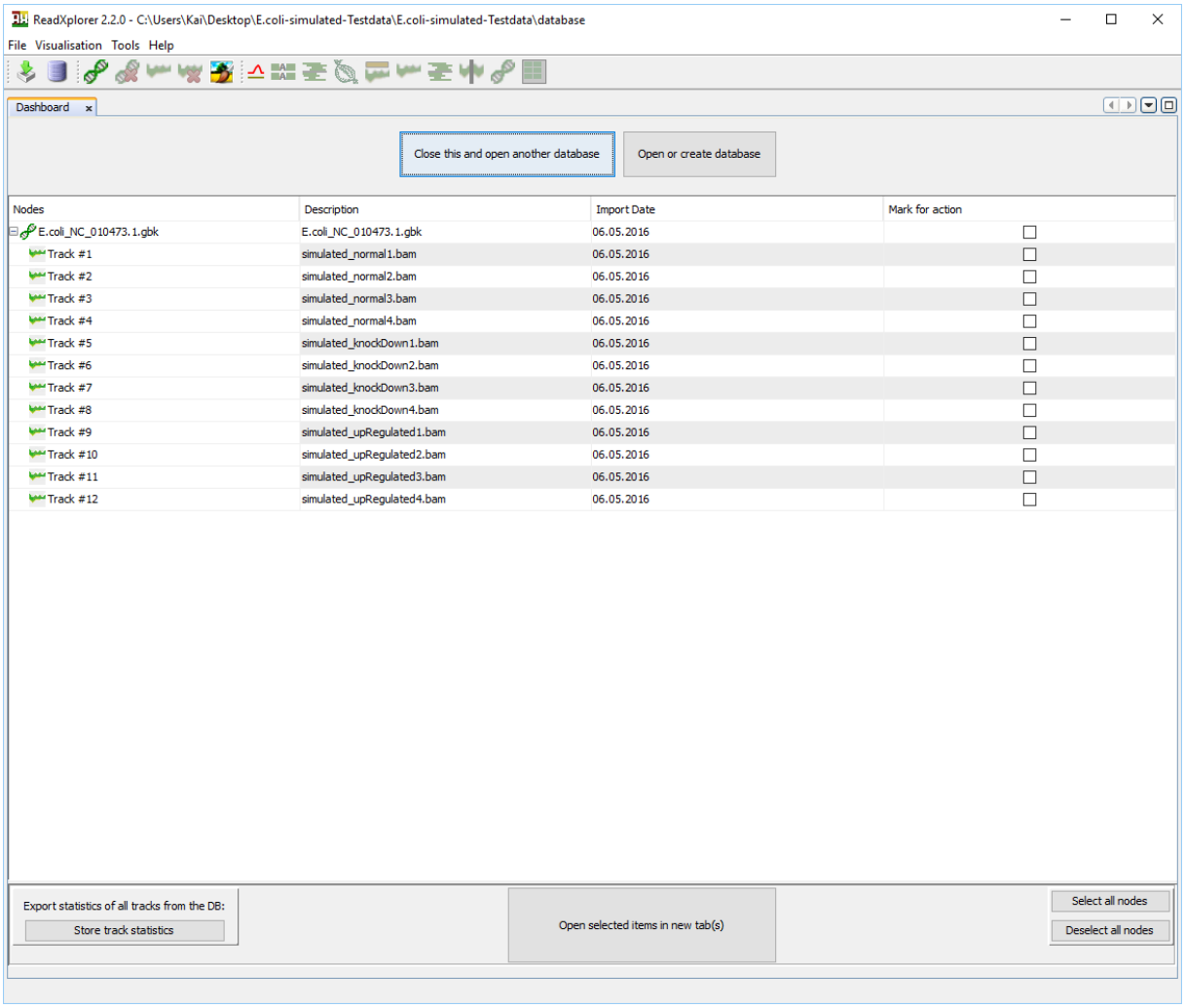
We recommend that you start with the test data-set provided on our homepage[3]. This data-set contains simulated RNA-Seq data from *E.coli*. Once downloaded you have to unzip it first.

When you start ReadXplorer for the first time it will look like this:



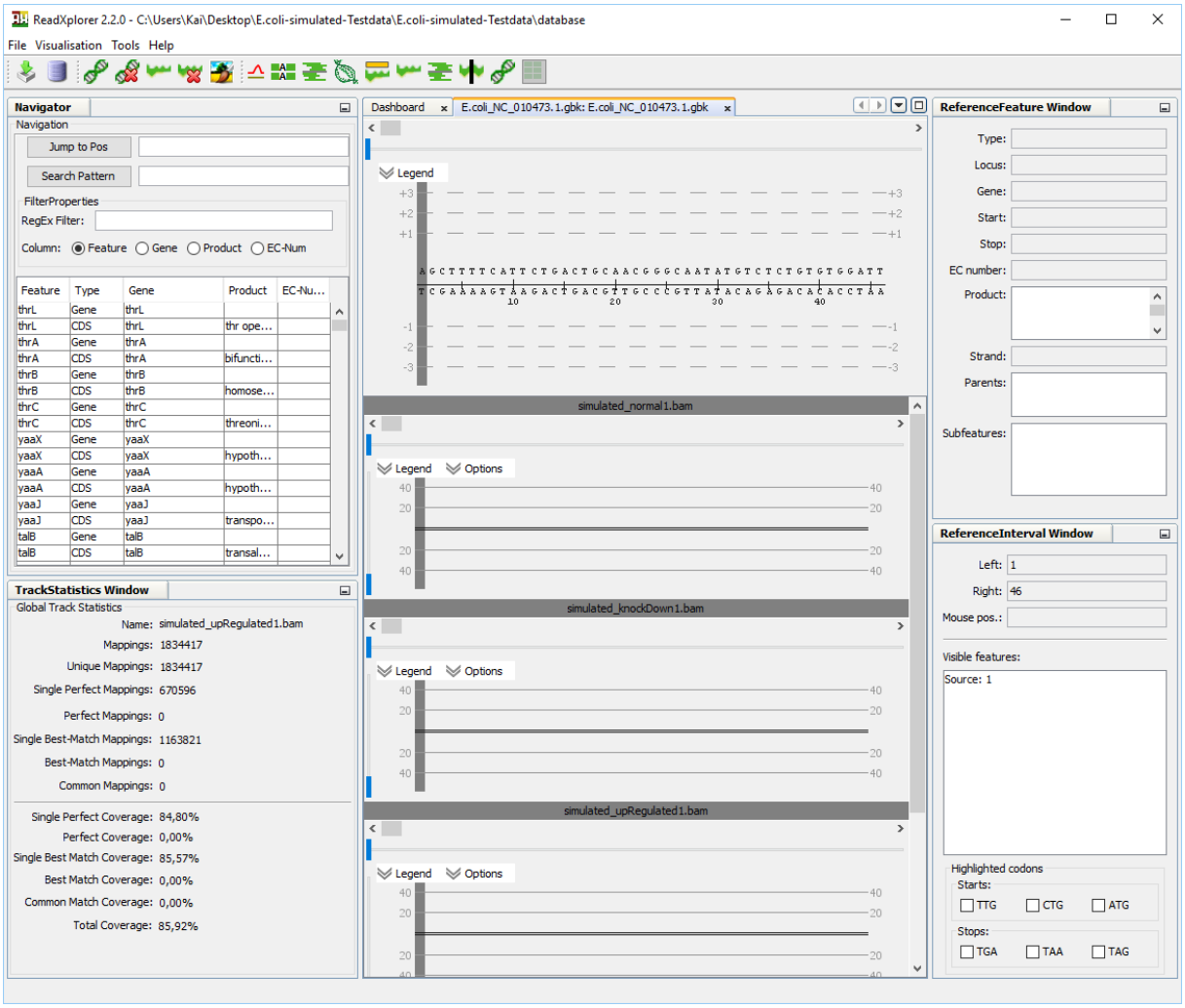
[3] ftp://ftp.cebitec.uni-bielefeld.de/pub/readxplorer_repo/E.coli-simulated-Testdata.zip

Select “Open existing database”, navigate to the folder where you unzipped the test-data set and open the database file (called “database” in this case). The view will now show you the different references and associated tracks in the database:



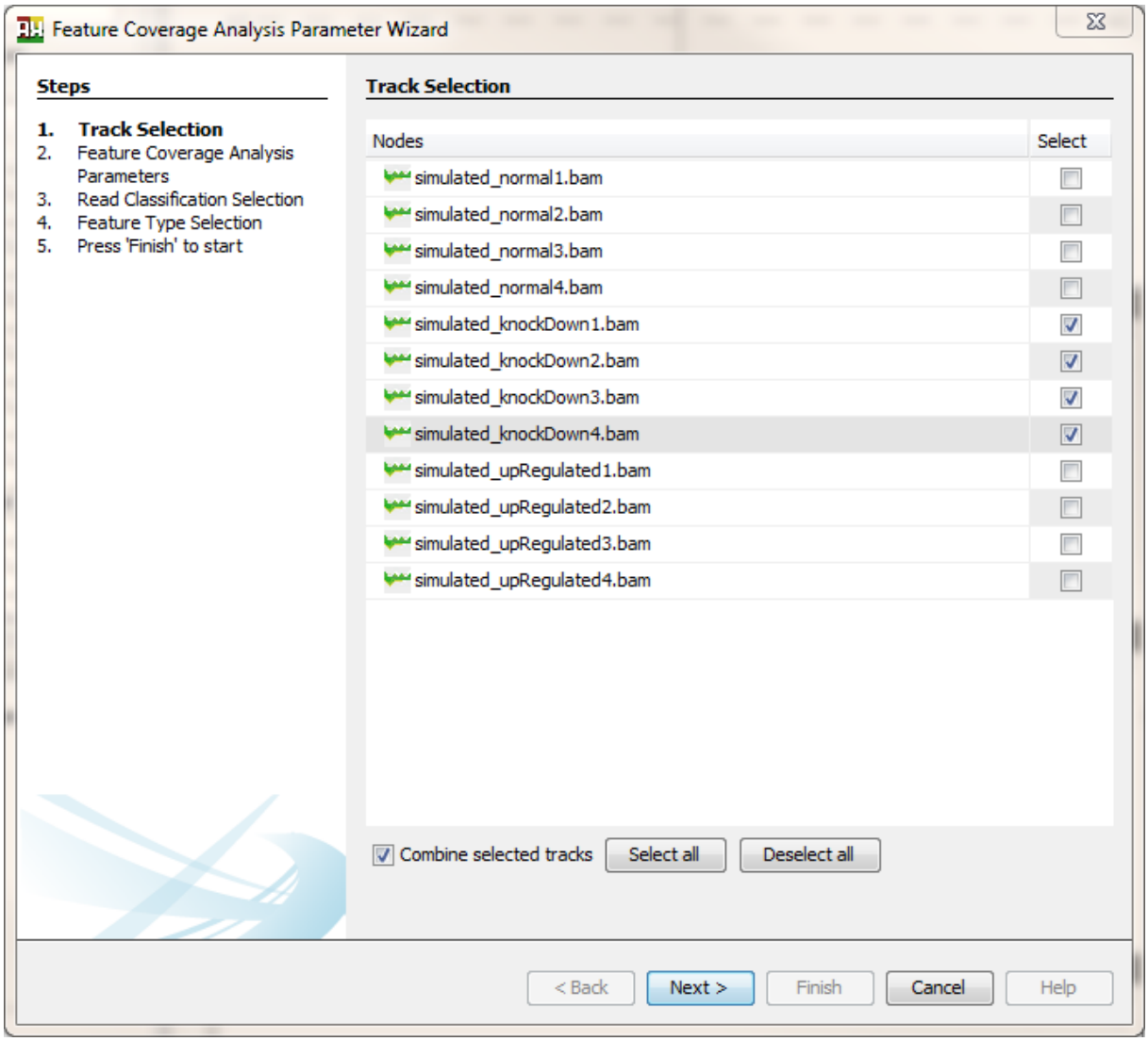
A database can hold multiple references and each reference can have multiple tracks associated to it. In this case, the database contains one reference and twelve associated tracks. Four tracks belong together; they are simulated replicates of different conditions one might find during an RNA-Seq experiment. All tracks with “normal” in their name represent RNA-Seq data generated under standard conditions (e.g. *E. coli* grown on normal media, at room temperature). In contrast the tracks with “knockDown” or “upRegulated" in their name contain data one might find when generating RNA-Seq data from non-standard conditions (e.g. different media used, heat stress etc.). In these tracks some genes are down or up regulated in comparison to the normal data-set. You can mark the tracks and open them using the “open selected items in new tab” button. We will mark and open one track of each condition.

The following tab will open:

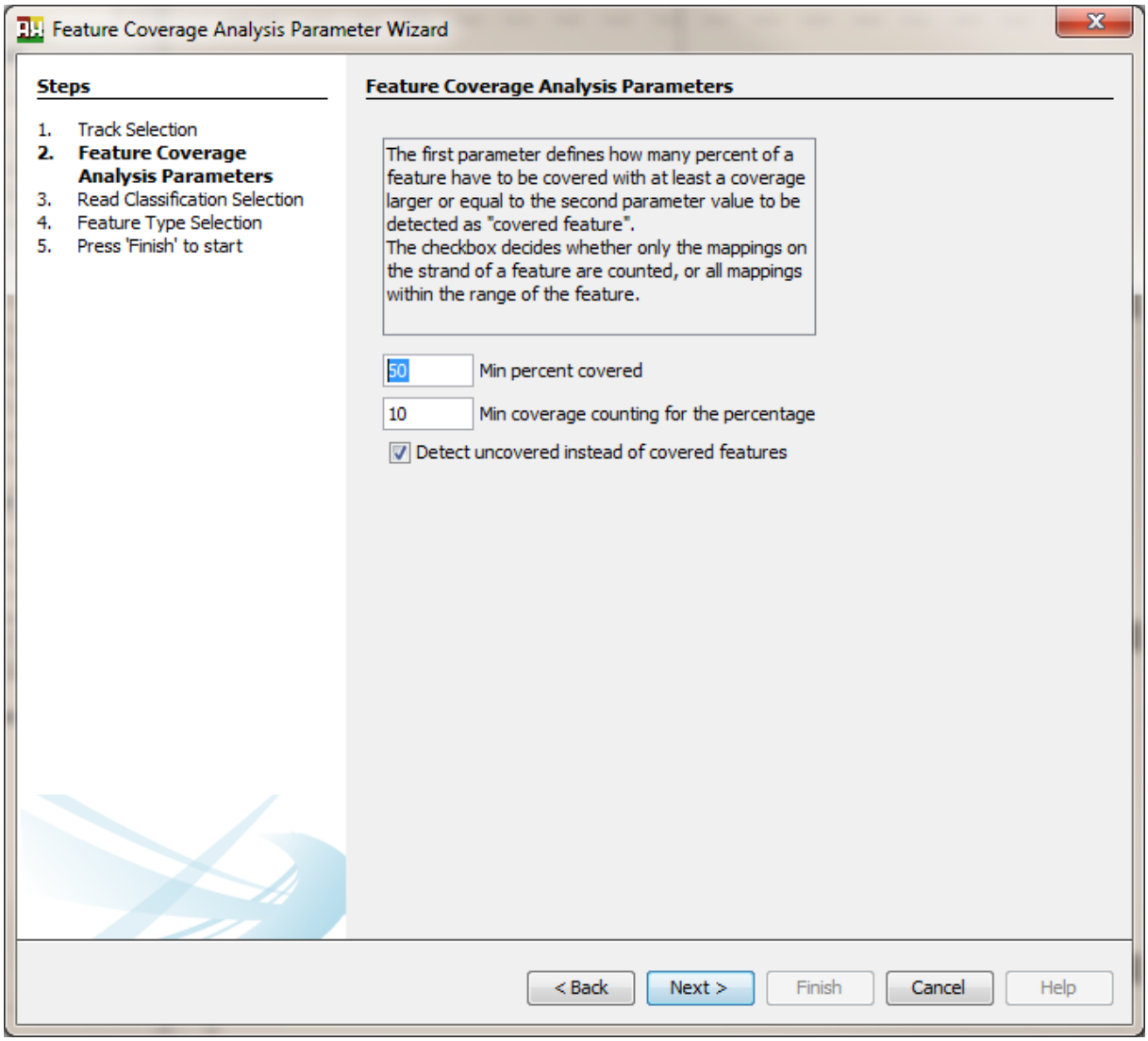


This is ReadXplorer's main viewing window. The different elements are described in the manual. Aside from the visualization ReadXplorer offers various analysis tools. As a first example we will use the "Feature Coverage Analysis" to look for uncovered coding sequences (CDS) in the data-set. All analysis functions are configured by a wizard. You can access all analysis functions from the "Tools" menu.

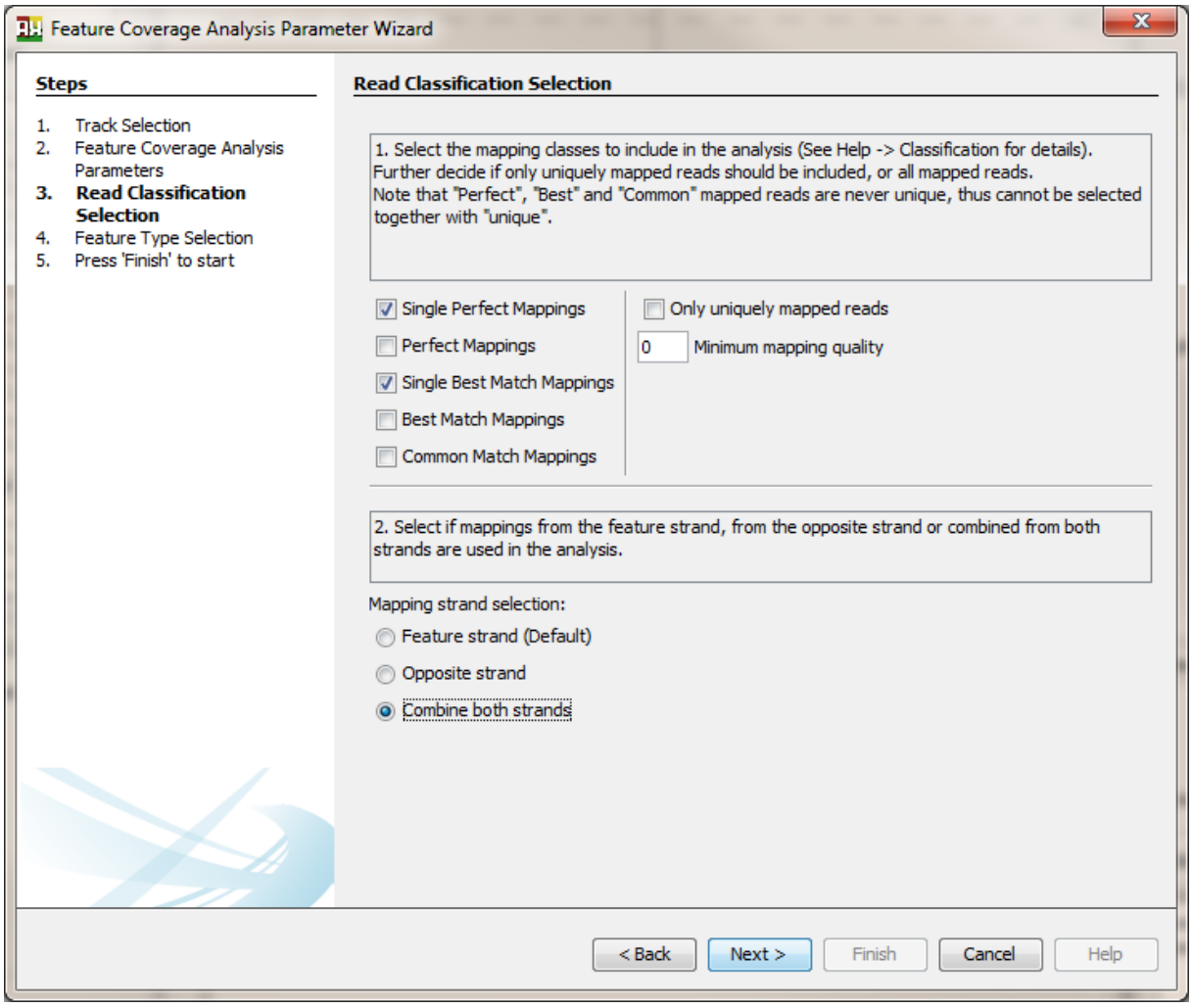
The first wizard panel asks which tracks should be included in the analysis. We select all “knockDown” tracks and additionally check the “Combine selected tracks” checkbox. When this box is checked, ReadXplorer will combine the read information from the selected tracks and treat them as if they were contained in one single track.



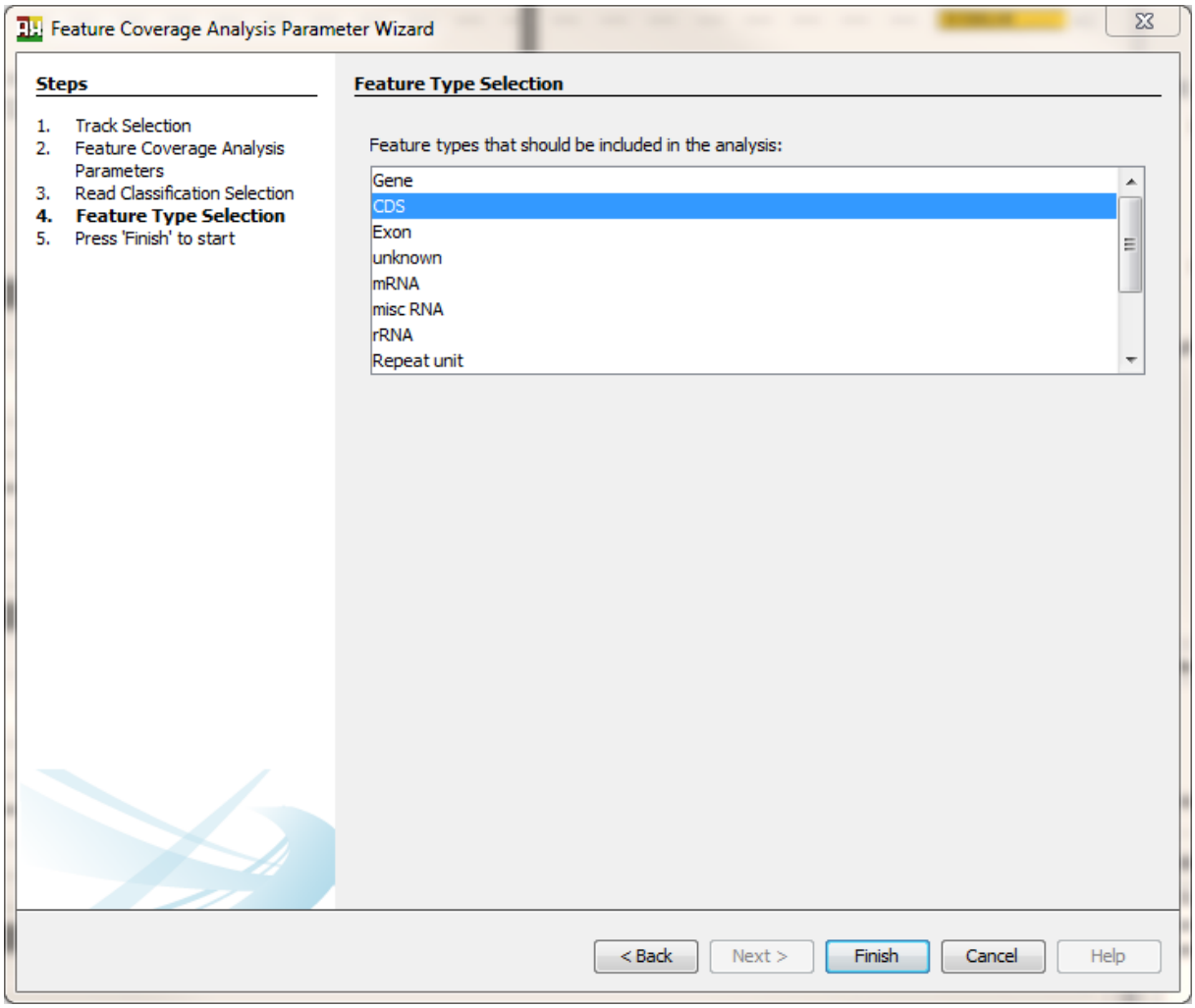
In the next panel the parameters for the analysis are set. As we want only uncovered CDS we check “Detect uncovered instead of covered features”, additionally we decrease the “Min percent covered” value to “50”.



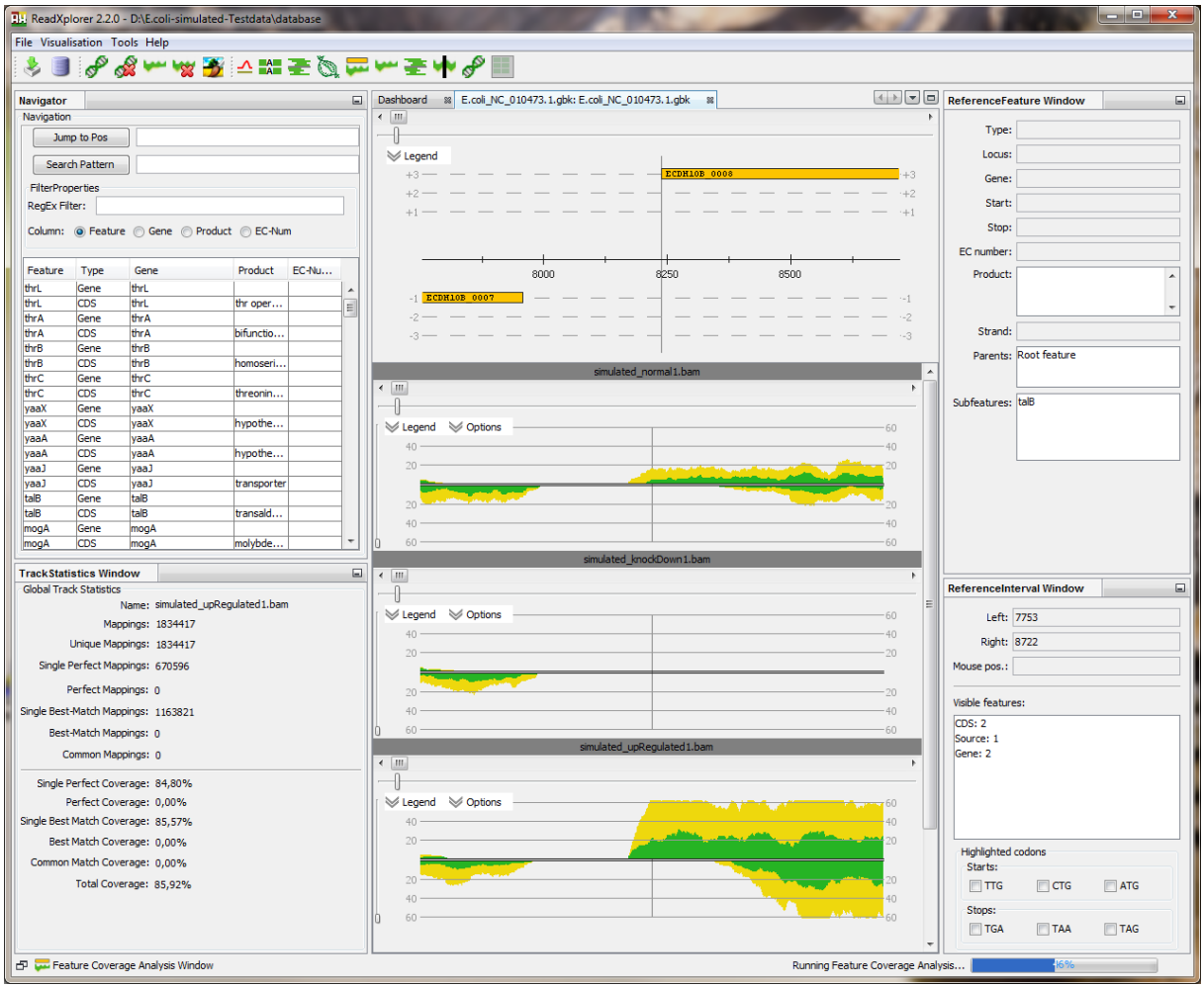
In the next panel we can select which read classes should go into the analysis. Please refer to the manual for a detailed explanation about the different read classes. We also select “Combine both strands”. This option is useful for data-sets that are not strand specific, as in such data-sets the reads can map to the forward or reverse strand arbitrarily regardless of their biological origin.



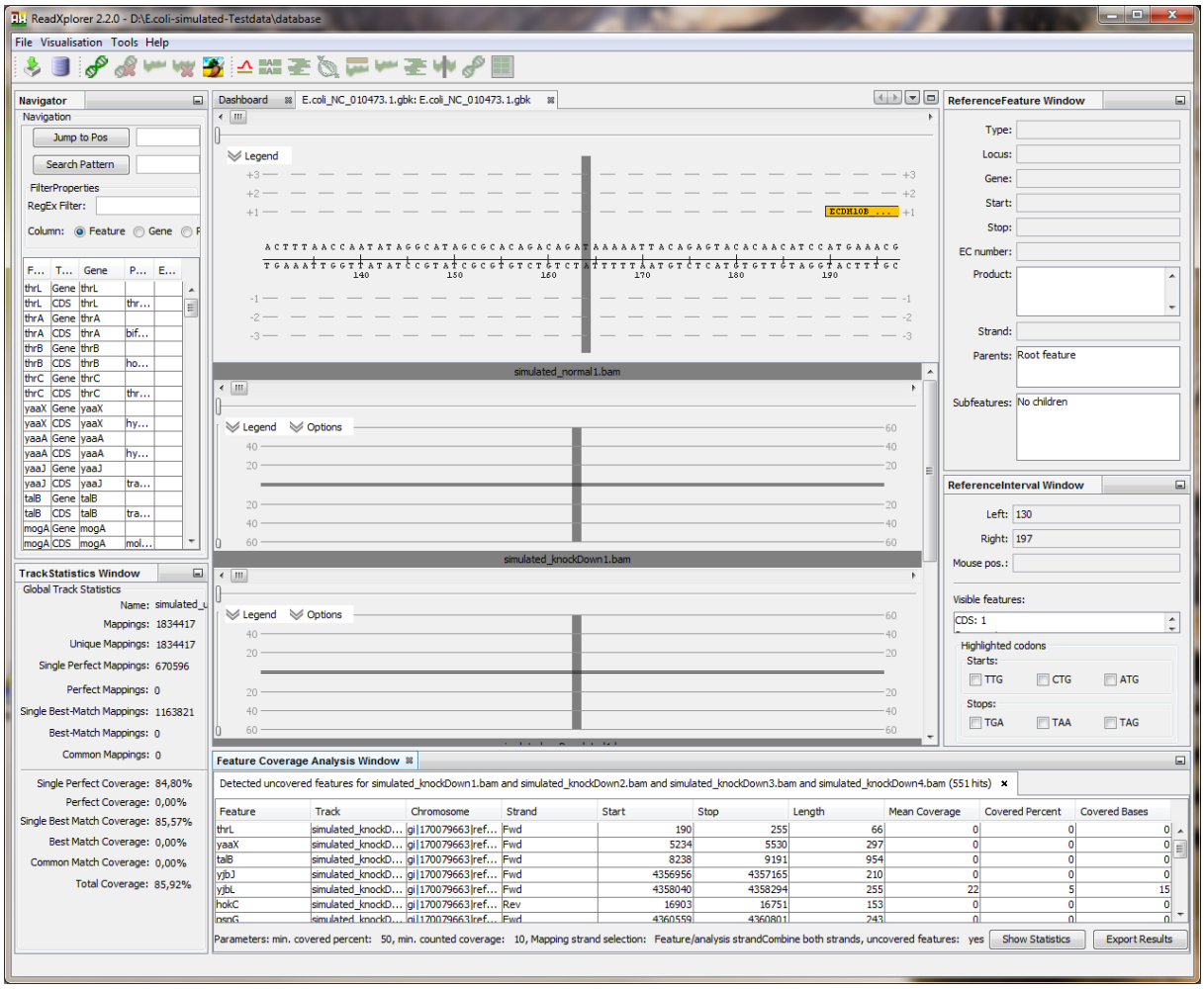
In the next step the annotation type the analysis should be applied to must be selected. As we intend to look at CDS we only select the “CDS” feature type. We the finished button is clicked the analysis will start.



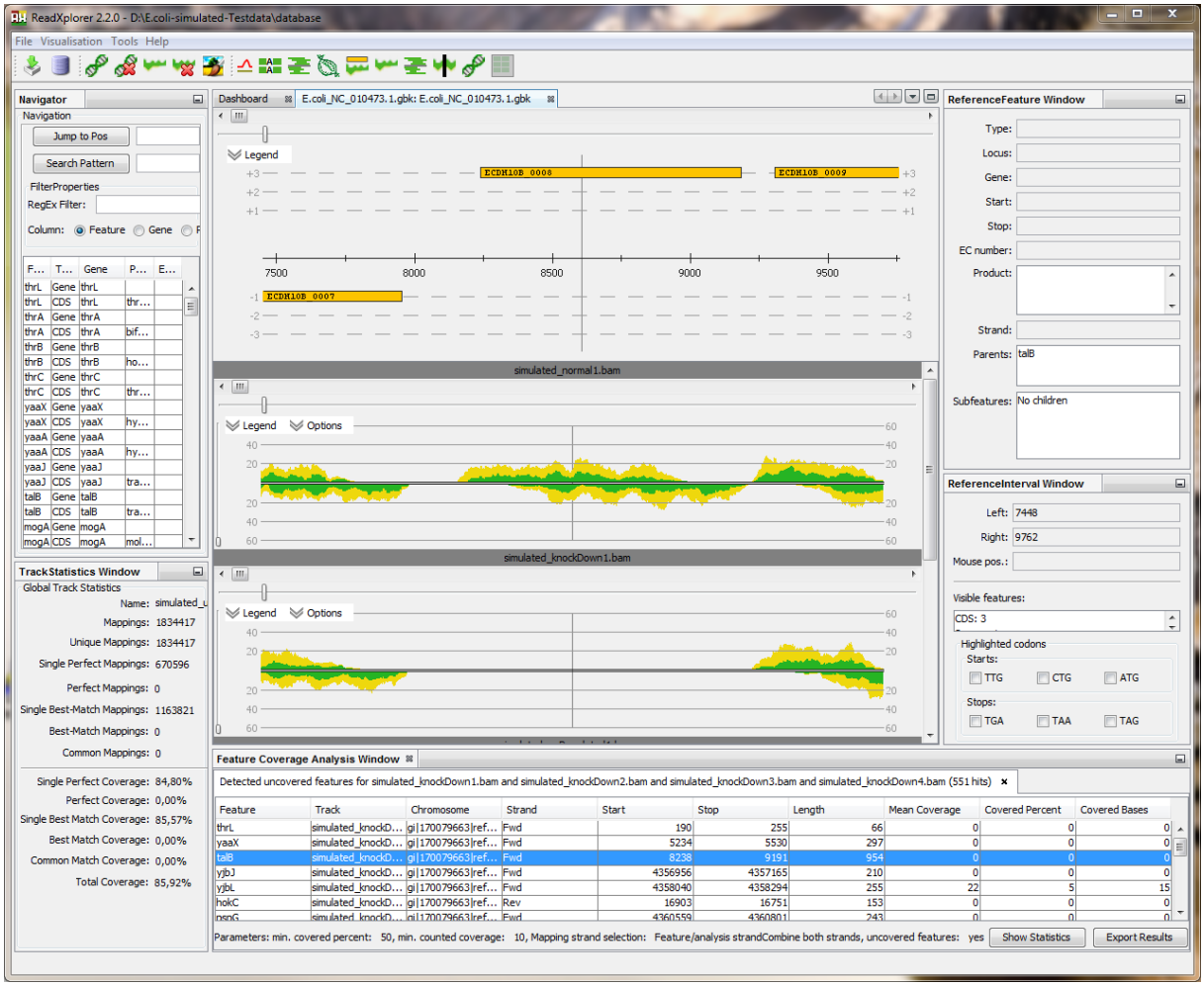
While the analysis is running you can continue using ReadXplorer normally. The status of the analysis is indicated by a progress bar in the lower right corner. You can also start additional analysis tasks and ReadXplorer will process them in parallel.



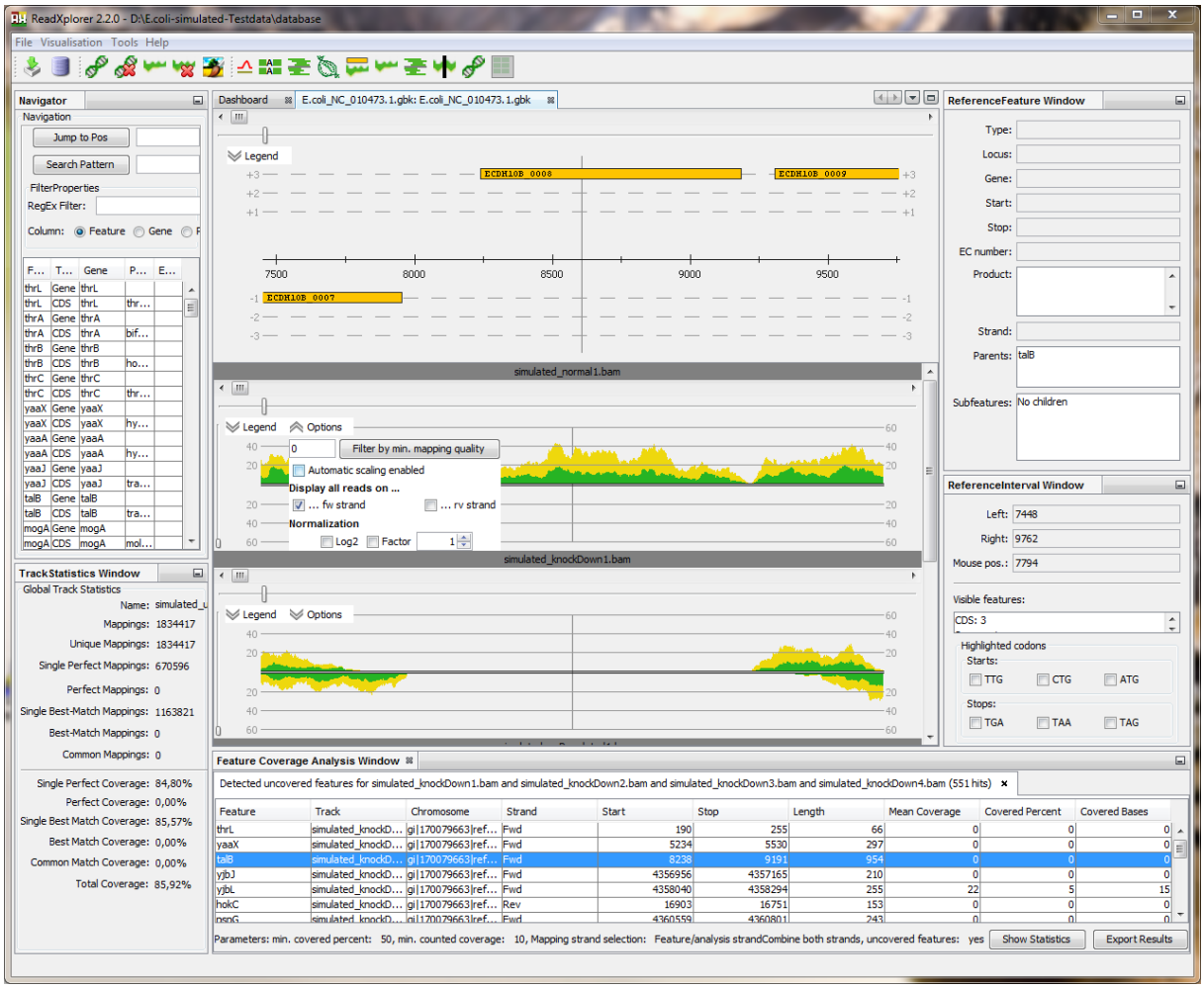
Once the analysis is finished a new window showing the results will open.



If you select an entry from the results table the viewer component will automatically jump to the start position of the associated annotation.

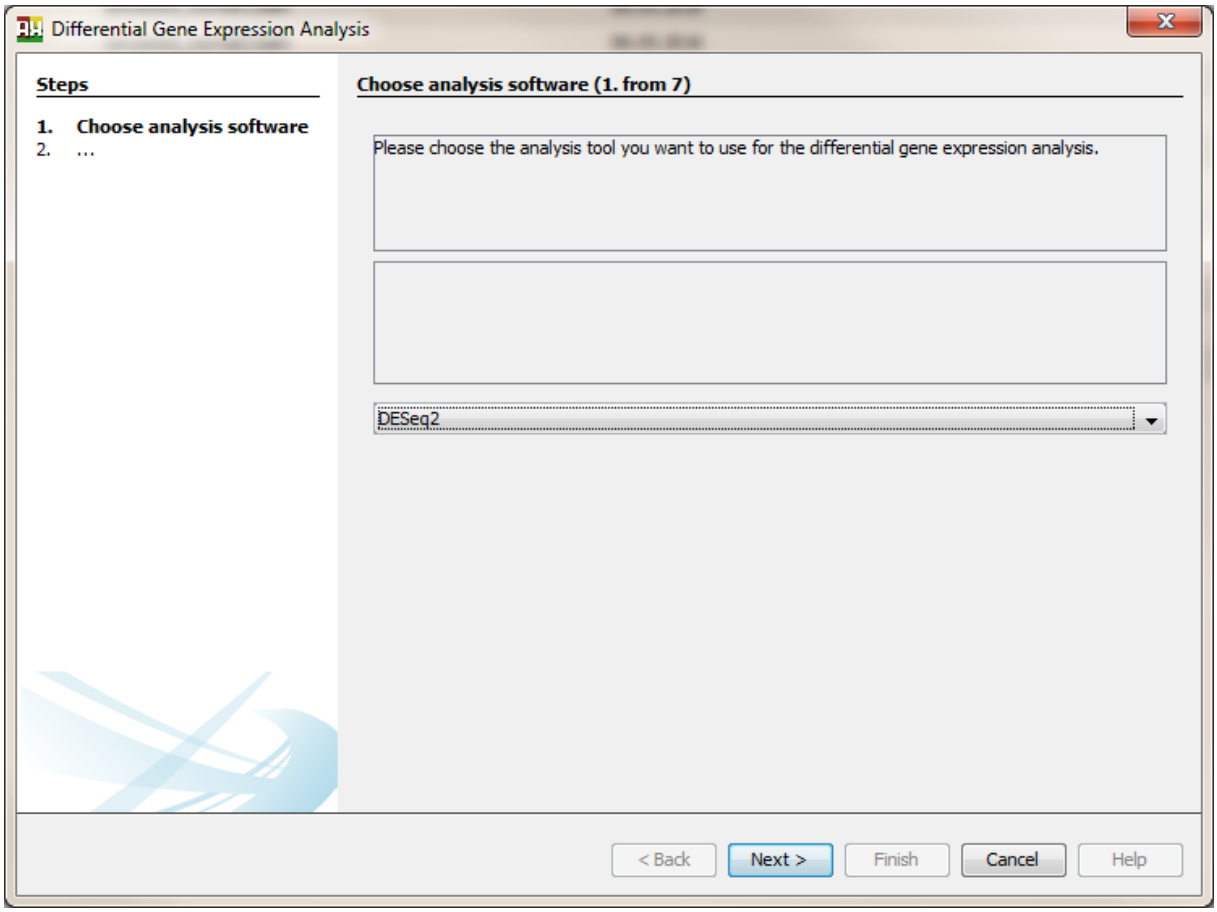


As the data-set is not strand specific you can adjust the view to show all reads on the fw strand.

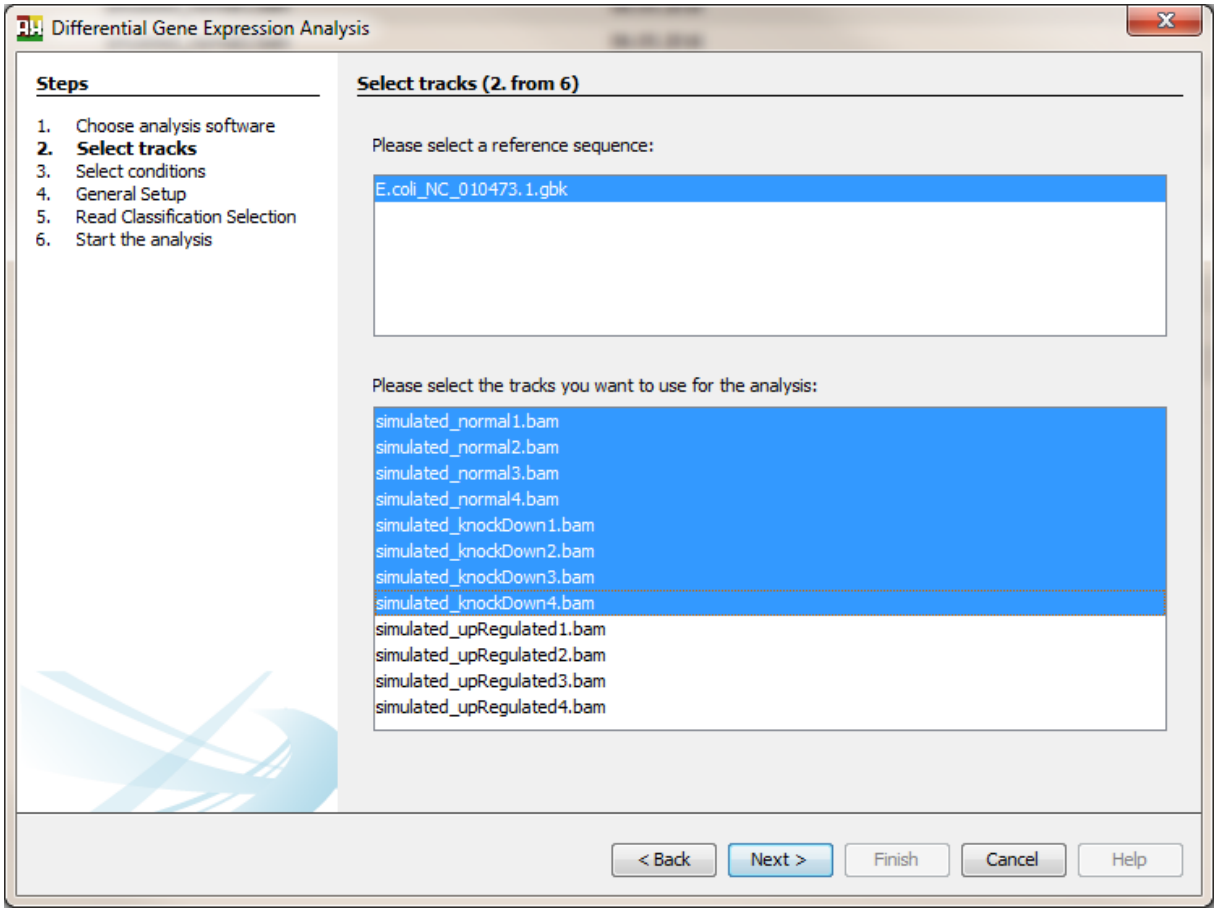


This is just one simple example of an integrated analysis function. For the next example we will use the more complex “Differential Gene Expression Analysis”, which relies on external tools that are not a native part of ReadXplorer. They need the statistical programming language GNU R to run and you need to set up the connection between ReadXplorer and GNU R once before you can start using the analysis. The setup options for different operating systems are described in the manual in detail. In general you have to visit the GNU R options panel within ReadXplorer at least once. The panel is located under “Tools” → “Options”. On Windows we highly recommend that you just select “Download and Install Gnu R”. This will automatically install everything that is needed to run the “Differential Gene Expression Analysis” on a windows machine. If you already have GNU R installed: Don’t worry, your existing installation will not be altered. For Linux operating systems and OS X you will have to install GNU R and the needed packages on your own as described in the manual. When you have done that go back to the GNU R options panel and set up the connection parameters that match your installation. If you are using a local R installation most of the time it should be sufficient to just change the radio button to “Manual”. Once the setup is complete you can start the “Differential Gene Expression Analysis” located in the tools menu.

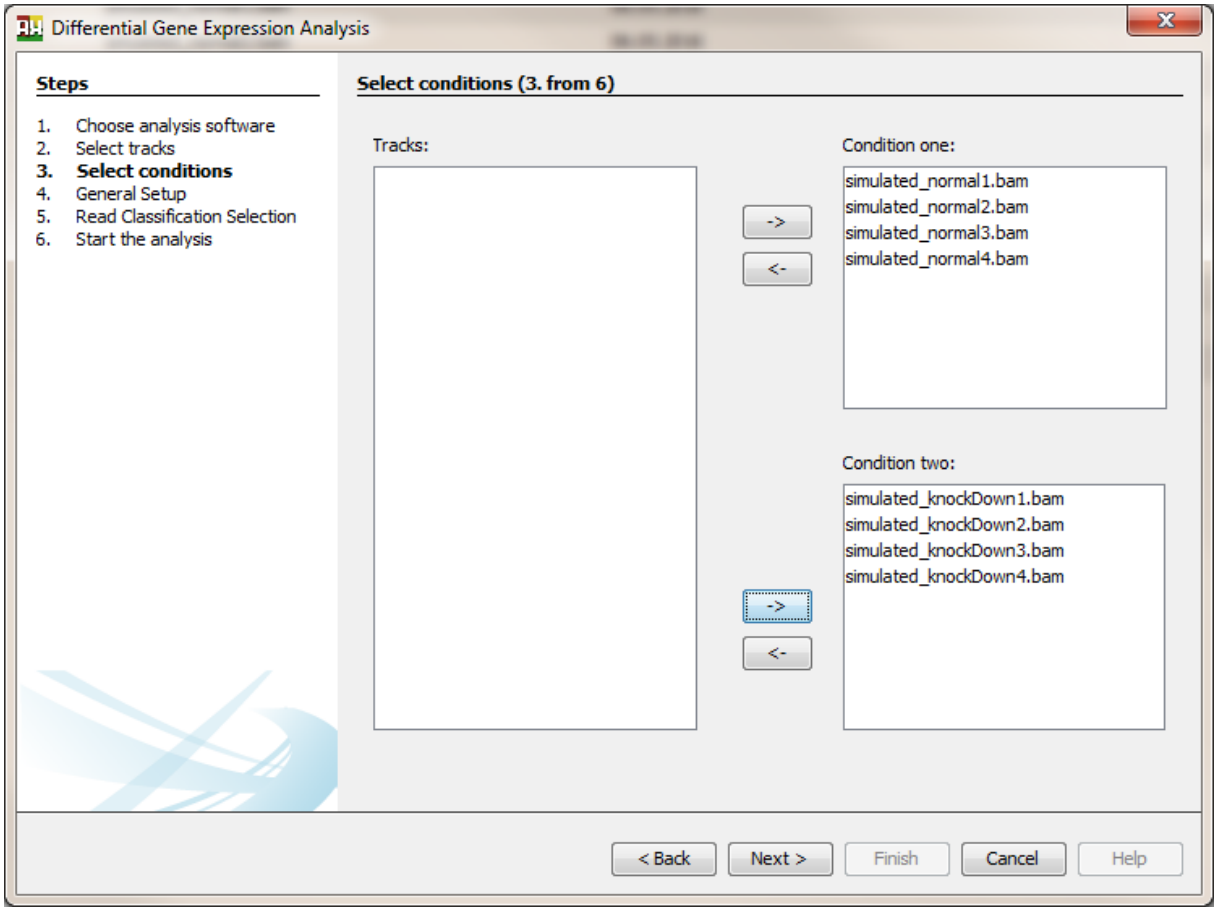
Once again a configuration wizard will collect the necessary information before the analysis is started. In the first panel you have to select the tool you want to use. For this tutorial we select “DESeq2”. Please refer to the manual for a short introduction of all the tools offered. Additionally, it is advisable to also read the publication of the tool you want to use.



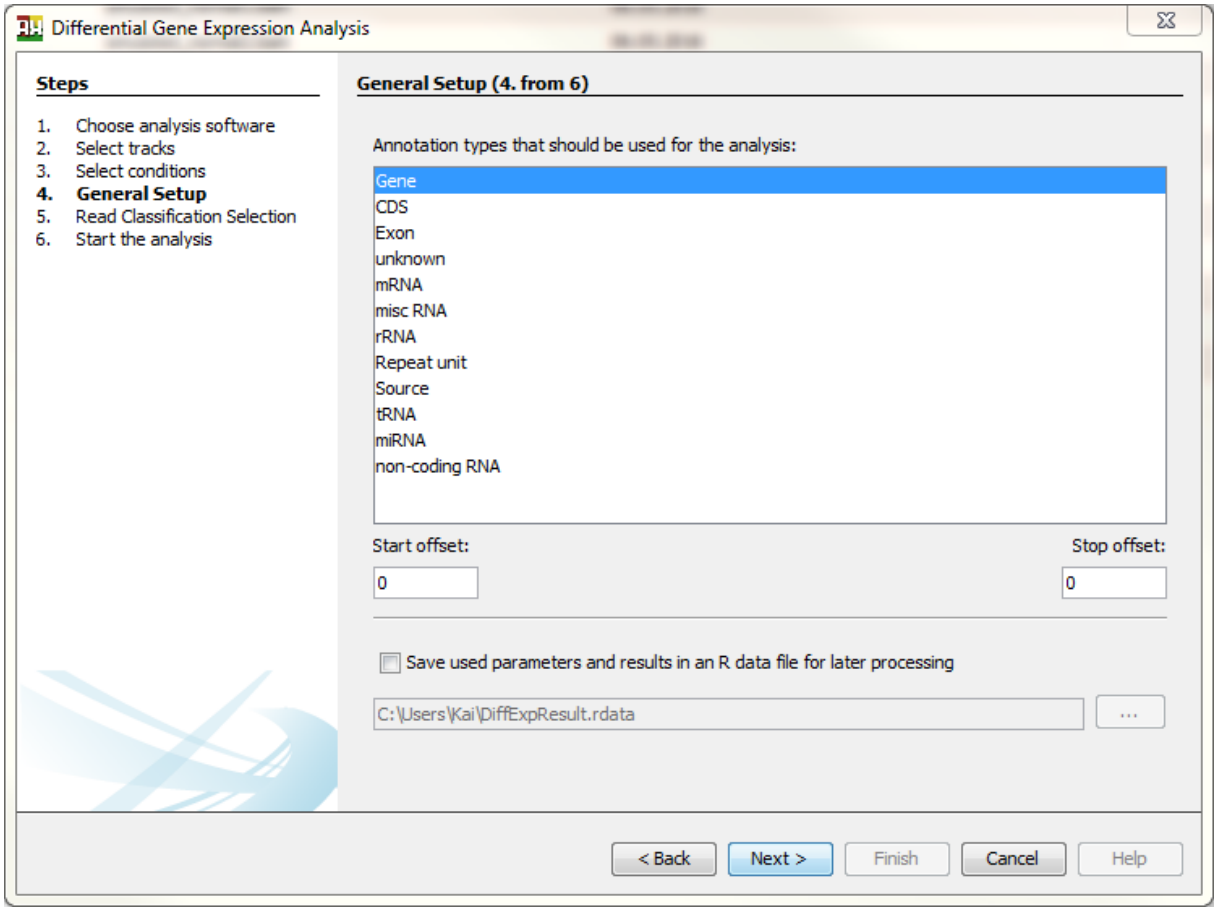
On the next panel select the tracks that should be included in the analysis. This time we select all “normal” and all “knockDown” tracks because we want to compare these to conditions.



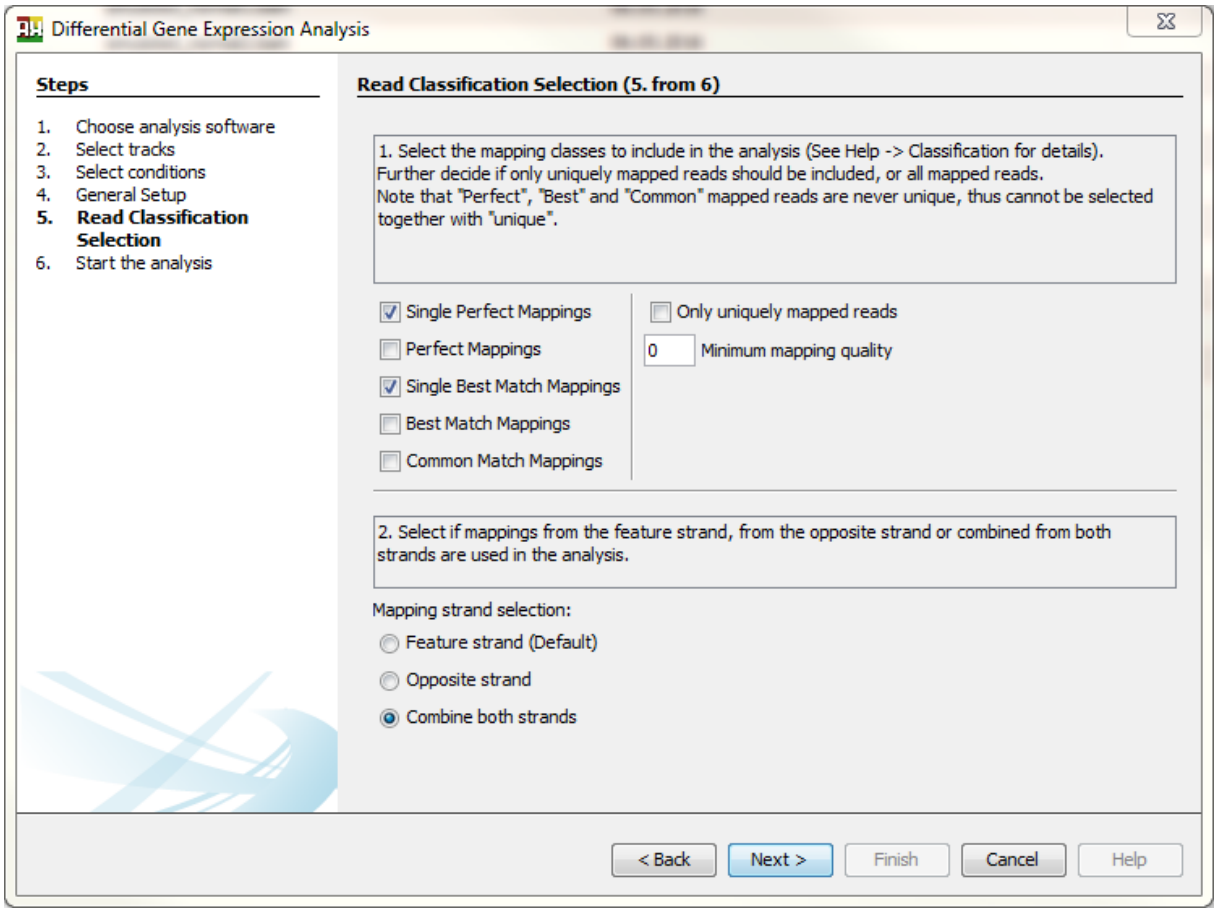
You must assign each selected track to a condition. Obviously, it only makes sense to assign all “normal” tracks to one condition and all “knockDown” tracks to the other.



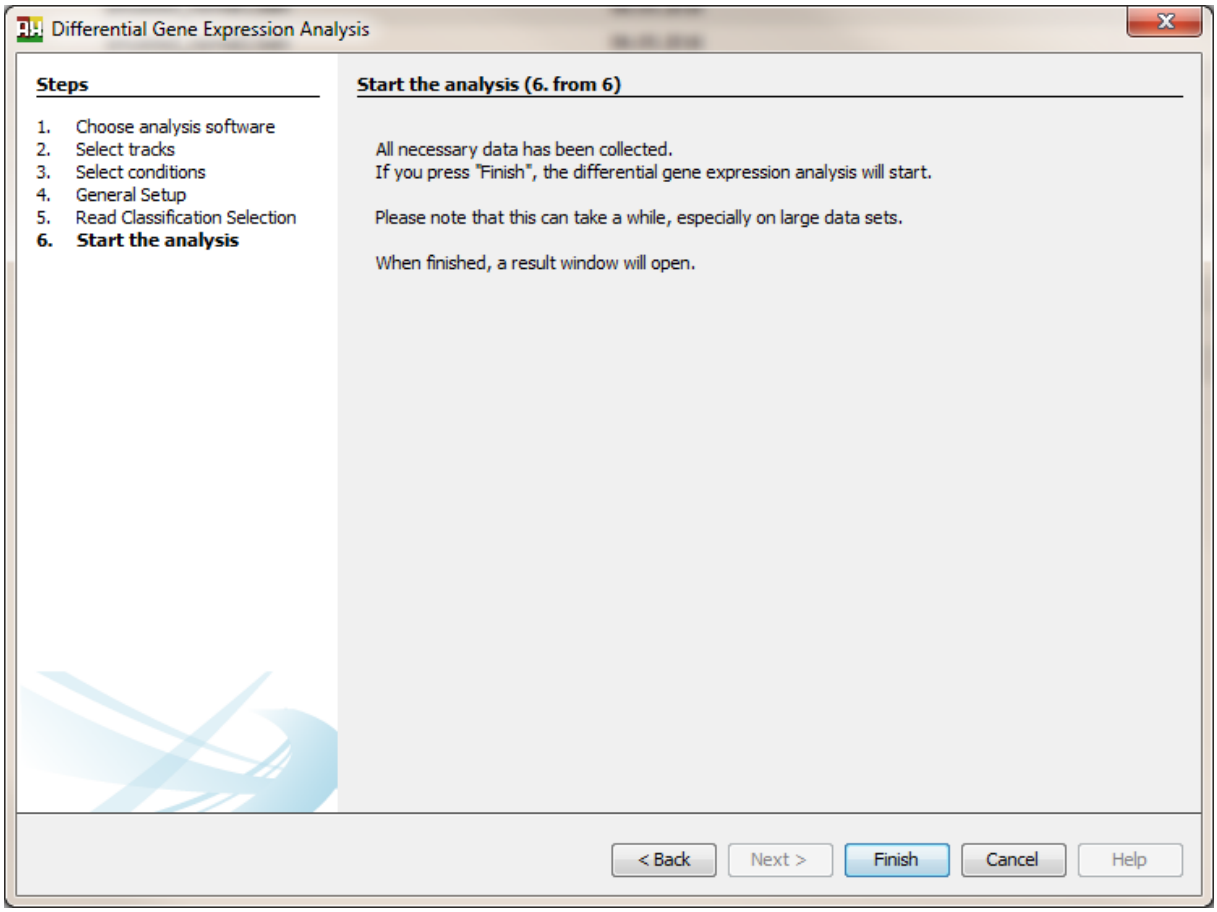
In the next panel the annotations that should be used for the analysis run must be selected. This time we choose “Gene”.



In the “Read classification” panel we change the radio button to “Combine both strands”.

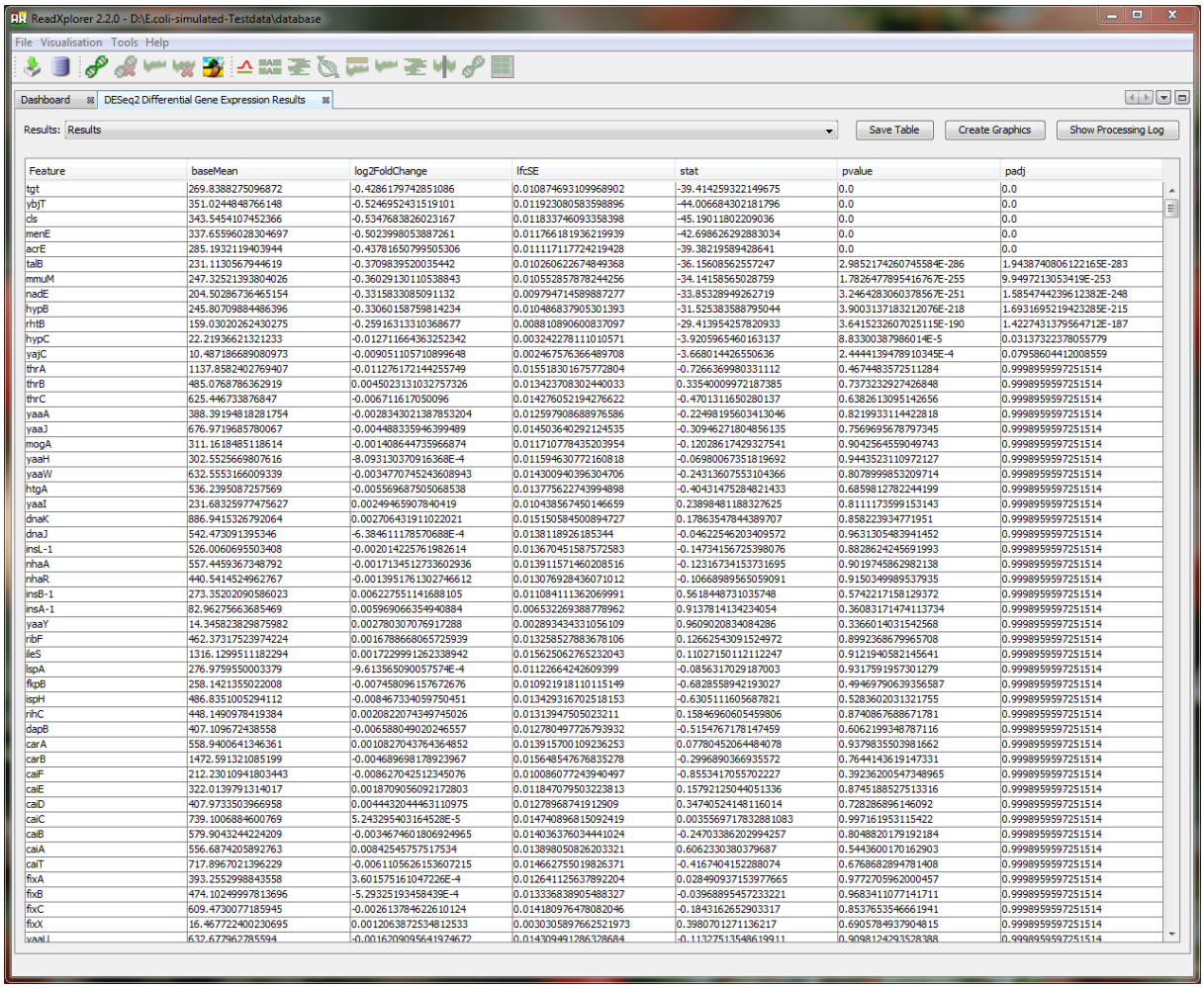


Finally we start the analysis with a click on “Finished”.



ReadXplorer will now count the reads for all genes of the reference and all selected tracks and convert them to a format DESeq2 can handle. The progress is indicated once again on the lower right corner.

Once DESeq2 is finished a new interactive panel will open showing the results. If you click on an entry the viewer will automatically bring you to the corresponding position. This offers an easy way to visually inspect the results generated by the analysis tool.

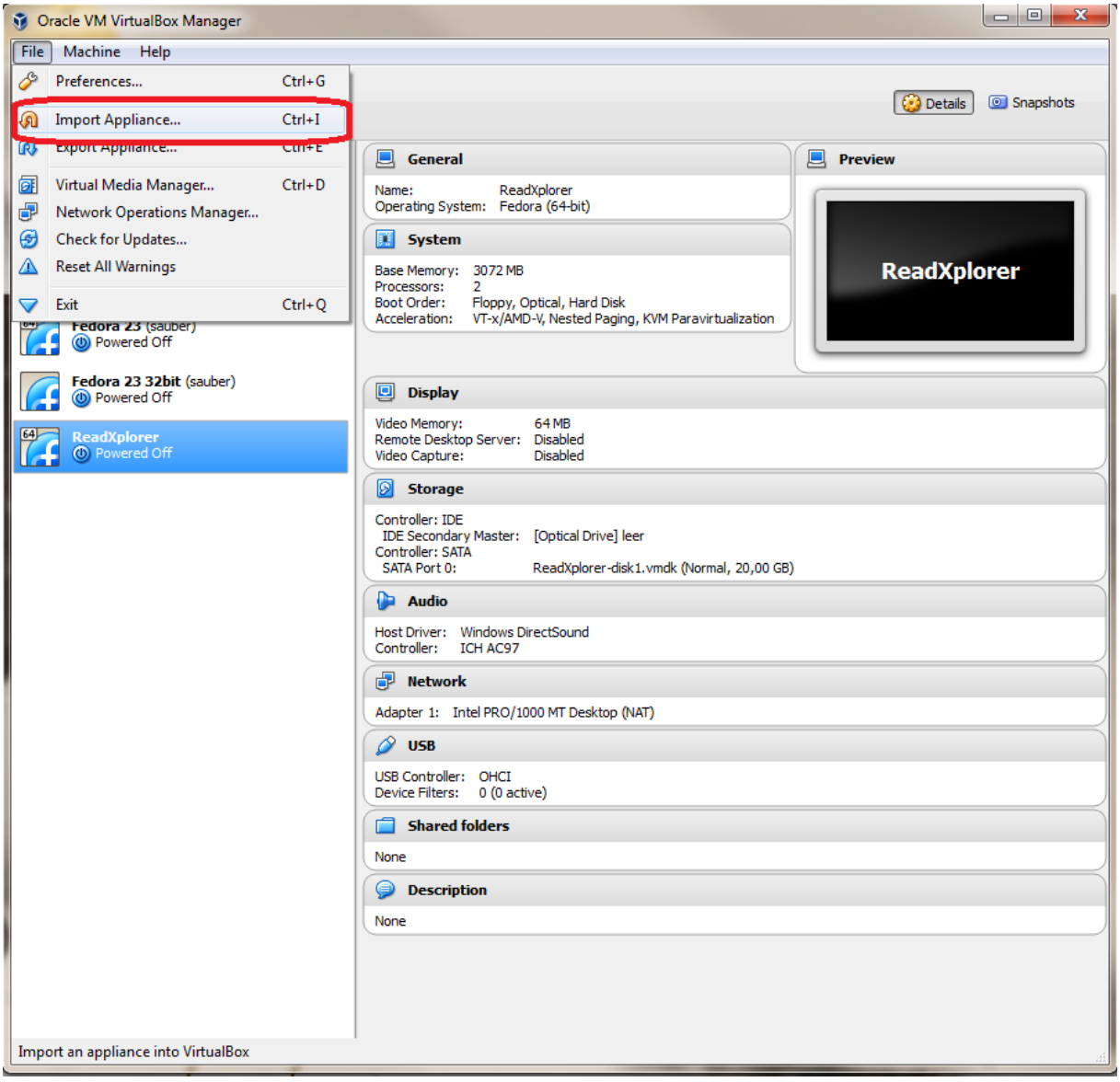


Feature	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
tgt	269.8388275096872	-0.42861797428510 86	0.010874693109968902	-39.414259322149675	0.0	0.0
ybjT	351.02448487661 48	-0.5246952431519101	0.011923080583598896	-44.006684302181796	0.0	0.0
cls	343.5454107452366	-0.5347683826023167	0.011833746093358398	-45.19011802209036	0.0	0.0
menE	337.65596028304697	-0.5023998053887261	0.011766181936219939	-42.698626292883034	0.0	0.0
acrE	285.1932119403944	-0.43781650799505306	0.011117117724219428	-39.38219589428641	0.0	0.0
talB	231.1130567944619	-0.3709839520035442	0.010260622674849368	-36.156085625572 47	2.9852174260745584E-286	1.9438740806122165E-283
mmuM	247.32521393804026	-0.3602913011053884 3	0.010552857878244256	-34.14158565028759	1.7826477895416767E-255	9.9497213053419E-253
nadE	204.50286736465154	-0.3315833085091132	0.009794714589887277	-33.853289492627 19	3.2464283060378567E-251	1.5854744239612382E-248
hypB	245.80709884486396	-0.3306015875981423 4	0.010486837905301393	-31.525383588795044	3.9003137183212076E-218	1.6931695219423285E-215
rhtB	159.03020262430275	-0.2591631331036867 7	0.008810890600837097	-29.413954257820933	3.6415232607025115E-190	1.4227431379564712E-187
hypC	22.21936621321233	-0.0127116643632523 42	0.003242278111010571	-3.9205965460163137	8.83300387986014E-5	0.03137322378055779
yajC	10.487186689080973	-0.0090511057108996 48	0.002467576366489708	-3.668014426550636	2.4444139478910345E-4	0.07958604412008559
thrA	1137.8582402769407	-0.0112761721442557 49	0.015518301675772804	-0.7266369980331112	0.4674483572511284	0.9988595597251514
thrB	485.0768786362919	0.0045023131032757326	0.013423708302440033	0.33540009972187385	0.7373232927426848	0.9988595597251514
thrC	625.446733876847	-0.0067116170500 96	0.014276052194276622	-0.4701311650280137	0.6382613095142656	0.9988595597251514
yaaA	388.39194818281754	-0.0028343021387853204	0.012597908688976586	-0.2249819560341 3046	0.8219933114422818	0.9988595597251514
yaaJ	676.9719685780067	-0.0044883359463994 89	0.014503640292124535	-0.3094627180485613 5	0.7569695678797345	0.9988595597251514
mogA	311.1618485118614	-0.0014086447359668 74	0.011710778435203954	-0.1202861742932754 1	0.9042564559049743	0.9988595597251514
yaaH	302.5525669807616	-8.0931303709163 68E-4	0.011594630772160818	-0.0698067351819692	0.9443523110972127	0.9988595597251514
yaaW	632.5553166009339	-0.0034770745243608943	0.014300940963047 06	-0.2431360755310 4366	0.8078999853209714	0.9988595597251514
htgA	536.2395087257569	-0.0055696875050 68538	0.013775622743994898	-0.4043147528482 1433	0.6859812782244199	0.9988595597251514
yaaI	231.68325977475627	0.0024946590784 0419	0.010438567450146659	0.2389848118832 7625	0.8111173599153143	0.9988595597251514
dnaK	886.9415326792064	0.0027064319110 22021	0.015150584500894727	0.1786354784438 9707	0.858223934771951	0.9988595597251514
dnaJ	542.473091395346	-6.3846111785706 88E-4	0.013811892618 5344	-0.0462254620340 9572	0.9631305483941452	0.9988595597251514
insL-1	526.0060695503408	-0.0020142257619 82614	0.013670451587572583	-0.1473415672539 8076	0.8828624245691993	0.9988595597251514
nhaA	557.4459367348792	-0.0017134512733 602936	0.013911571460208516	-0.1231673415373 1695	0.9019745862982138	0.9988595597251514
nhaR	440.5414524962767	-0.0013951761302 746612	0.013076928436071012	-0.1066898956505 9091	0.9150349989537935	0.9988595597251514
insB-1	273.35202090586023	0.0062275511416 88105	0.011084111362069991	0.5618448731035748	0.5742217158129372	0.9988595597251514
insA-1	82.96275663685469	0.0059690663549 40884	0.006532269388778962	0.9137814134234054	0.3608317147411373 4	0.9988595597251514
yaaY	14.345823829875982	0.0027803070769 17288	0.002893434331056 109	0.9690920834084286	0.3366014031542568	0.9988595597251514
ribF	462.37317523974224	0.0016788668065725939	0.013258527883678106	0.1266254309152497 2	0.8992368679965708	0.9988595597251514
ileS	1316.1299511182294	0.0017229991262338942	0.015625062765232043	0.1102715011211 2447	0.9121940582145641	0.9988595597251514
ispA	276.9759550003379	-9.6135650900575 74E-4	0.011226642426099 99	-0.0856317029187003	0.9317591957301279	0.9988595597251514
fkpB	258.1421355022008	-0.0074580961576 72676	0.010921918110115149	-0.6828558942193027	0.4946979063935 6587	0.9988595597251514
ispH	486.8351005294112	-0.0084673405975 0451	0.013429316702518153	-0.6305111605687821	0.5283602031321755	0.9988595597251514
nhC	448.1490978419384	0.0020822074349 745026	0.013139475050232 11	0.1584696060545 9806	0.8740867688671781	0.9988595597251514
dapB	407.109672438558	-0.0065880490202 46557	0.012780497726793932	-0.5154767178147459	0.6062199348787116	0.9988595597251514
carA	558.9400641346361	0.0010827043764364852	0.013915700109236253	0.0778045206448 4078	0.9379835503981662	0.9988595597251514
carB	1472.591321085199	-0.0046896981789239 67	0.015648547678352 78	-0.2996890366935572	0.7644143619147331	0.9988595597251514
caiF	212.23010941803443	-0.0086270425123 45076	0.010086077243940497	-0.8553417055702227	0.3923620054734 8965	0.9988595597251514
caiE	322.0139791314017	0.0018709056092172803	0.011847079503223813	0.1579212504405 1336	0.8745188527513316	0.9988595597251514
caiD	407.9733503966958	0.0044432044463 110975	0.012789687419129 09	0.3474052414811 6014	0.7282868961460 92	0.9988595597251514
caiC	739.1006884600769	5.243295403164528E-5	0.014740896815092419	0.0035569717832881083	0.9971619531154 22	0.9988595597251514
caiB	579.9043244224209	-0.0034674601806924965	0.014036376034441024	-0.2470338620299 4257	0.8048820179192184	0.9988595597251514
caiA	556.6874205892763	0.0084255757571753 4	0.013898050826203321	0.6062330380379687	0.5443600170162903	0.9988595597251514
caiT	717.8967021396229	-0.0061105626153607215	0.014662755019826371	-0.4167404152288074	0.6768682894781408	0.9988595597251514
fixA	393.2552998843558	3.601575161047226E-4	0.012641125637892204	0.0284909371539776 65	0.9772705962000457	0.9988595597251514
fixB	474.1024999781396	-5.2932519345843 9E-4	0.013336838905488327	-0.0396889545723 3221	0.9683411077141711	0.9988595597251514
fixC	609.4730077185945	-0.0026137846226 10124	0.014180976478082046	-0.1843162652903317	0.8537653546661941	0.9988595597251514
fixX	16.467722400230695	0.0012063872534 812533	0.003030589766252197 3	0.3980701271136217	0.6905784937904815	0.9988595597251514
yaaU	632.677962785594	-0.0016209095641974672	0.014309491286328684	-0.1132751354861991 1	0.9098124293528388	0.9988595597251514

6. The ReadXplorer VirtualBox Image

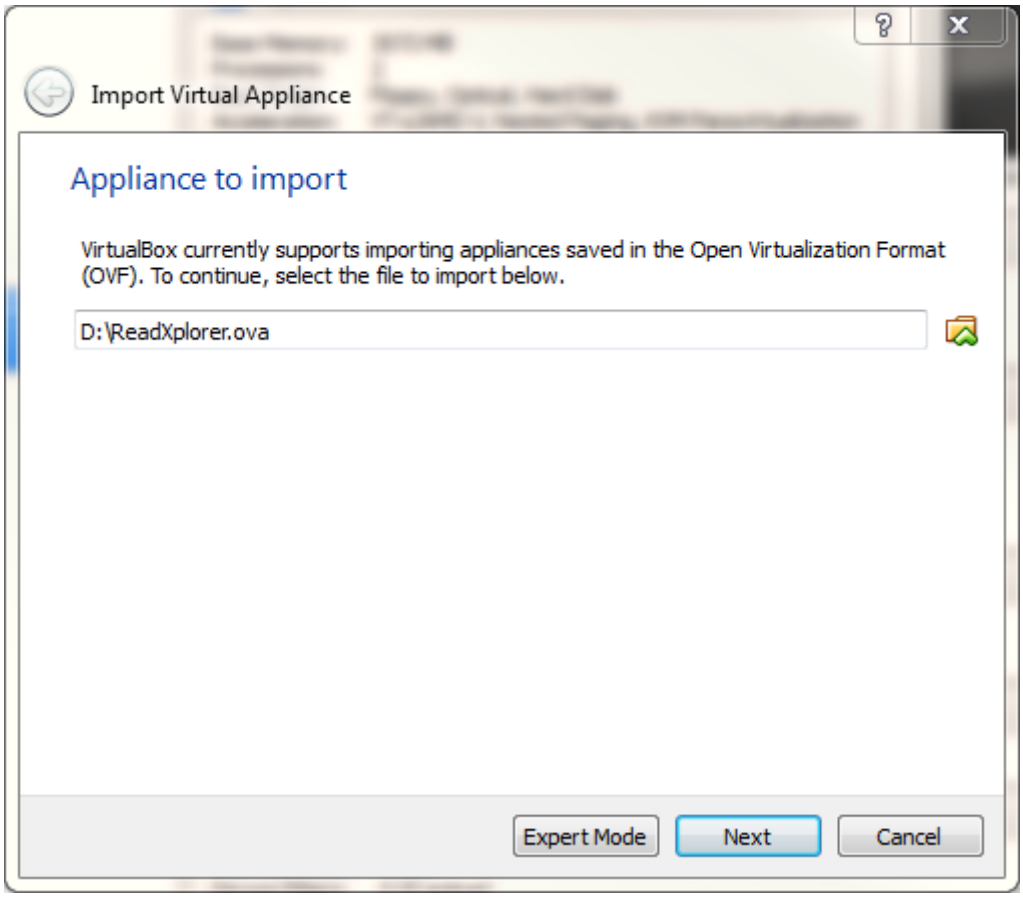
Virtualization offers the ability to emulate a complete computer on top of your already running operating system. This “virtual computer” is independent of your operating system and as long as the virtualization software is available for your already running operating system you can execute the “virtual computer”. This concept of virtualization makes it possible for us to prepare a “virtual computer” with all the necessary software installed to run ReadXplorer. You only need to install the version of the virtualization software matching your operating system and import the image of the “virtual computer” that we offer.

As virtualization software we choose VirtualBox[4]. It is available for most operating systems free of charge. Go to the download page[5] and download and install the version appropriate for you. Once the installation is finished go to our homepage and download the “ReadXplorer VirtualBox image”. Be aware that the download is quite large (~ 3.7 GB). When the download is finished, open VirtualBox and go to “File” -> “Import Appliance…”.

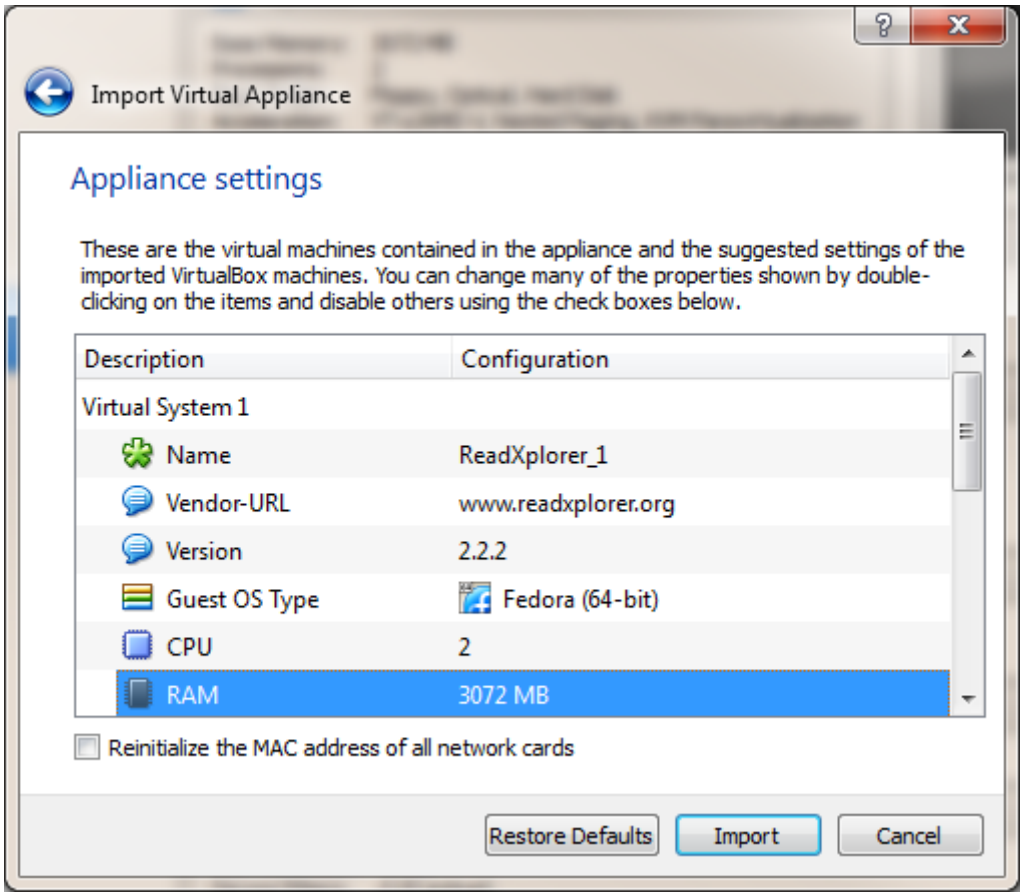


[4] https://www.virtualbox.org/
[5] https://www.virtualbox.org/wiki/Downloads

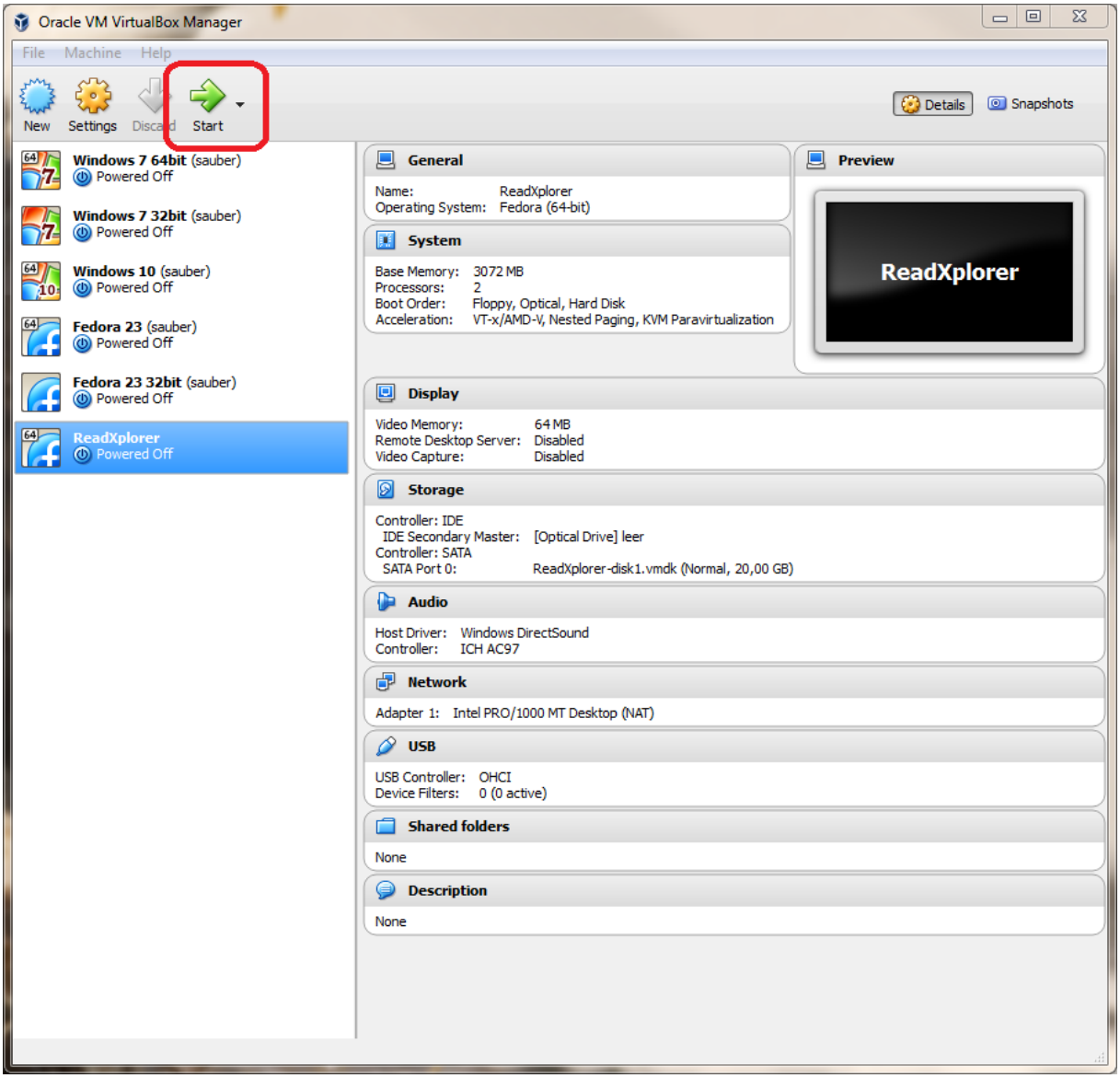
Select the location of the downloaded image file and continue by selecting “Next”:



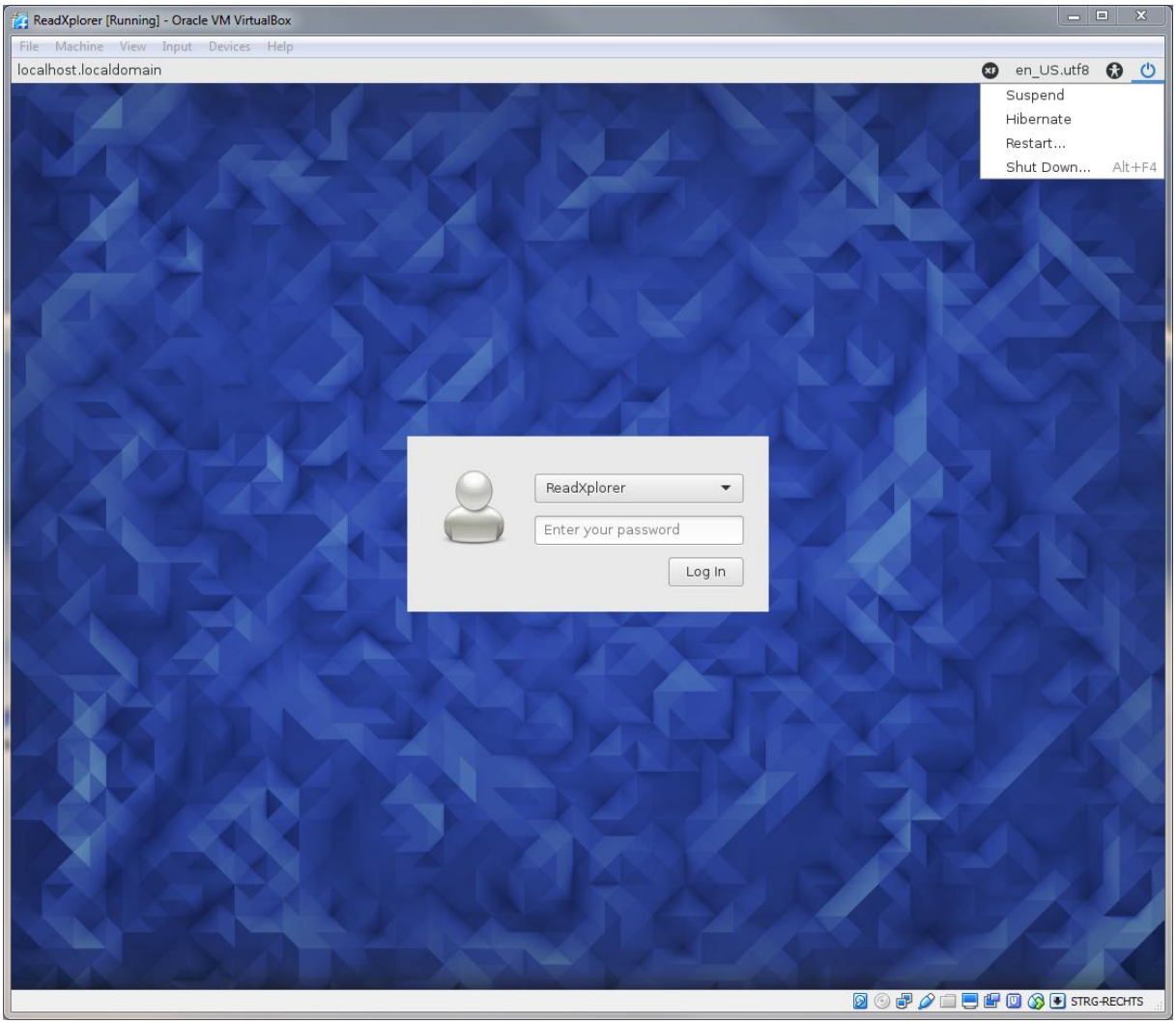
Adjust the “CPU” and “RAM” settings if necessary. The settings below are the standard settings chosen for a Laptop with a Quad-core CPU and 8 GB RAM. The value for RAM should be at least 2 GB smaller than the total amount of RAM in your computer. After adjusting the setting start the import process by selecting “Import”.



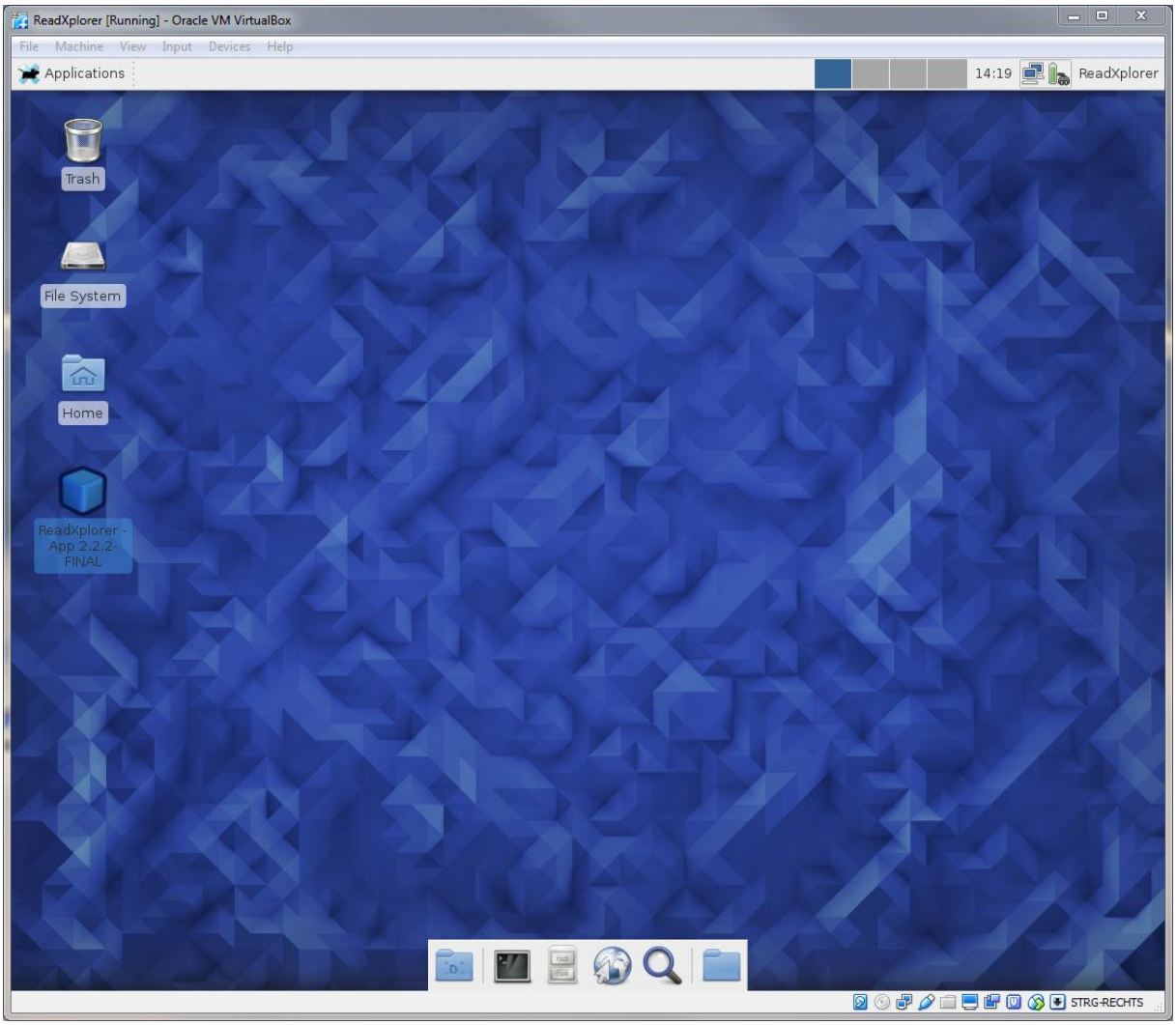
When the import is finished a new virtual machine called “ReadXplorer” will be visible. Select the virtual machine with one left klick and start it by selection “Start”.



The virtual machine will now boot a Fedora Linux in version 23. After booting a login screen will appear. The default user is called “readxplorer”. There is no password set for this user so you can login by simply selecting “Log In”.



You can launch ReadXplorer directly via the link located on the desktop.

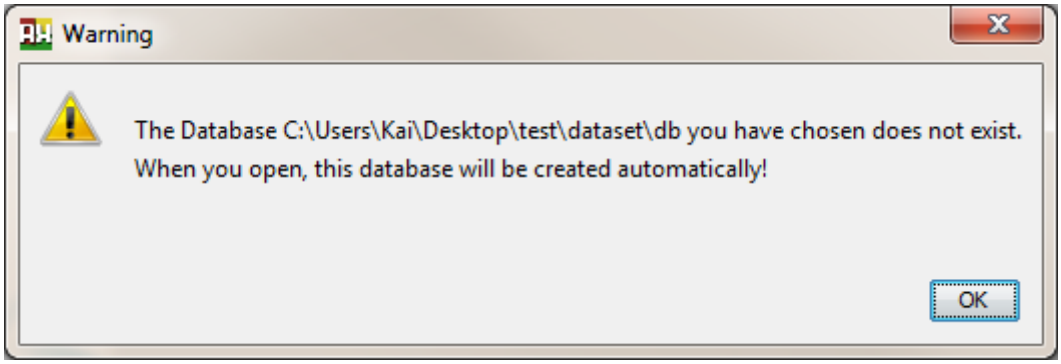


The virtual machine comes with a pre-installed GNU R installation. The *E.coli* test data-set is located in the home directory of the “readxplorer” user. This means that everything is already set-up and you can directly start using ReadXplorer and explore its functionality. If you need root access for some reason the password is simply “rx”.

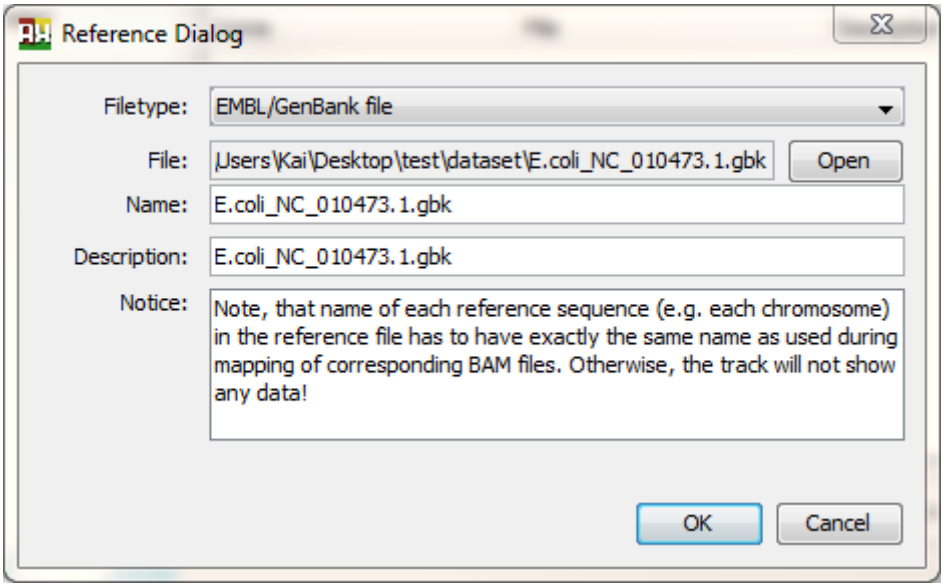
7. How can I use my own data with ReadXplorer?

ReadXplorer uses a database to maintain all important information it needs to visualize the mapping data. In order to use your own data you must import it first into a ReadXplorer database. The data you need is described in the section “What input data is needed?”.

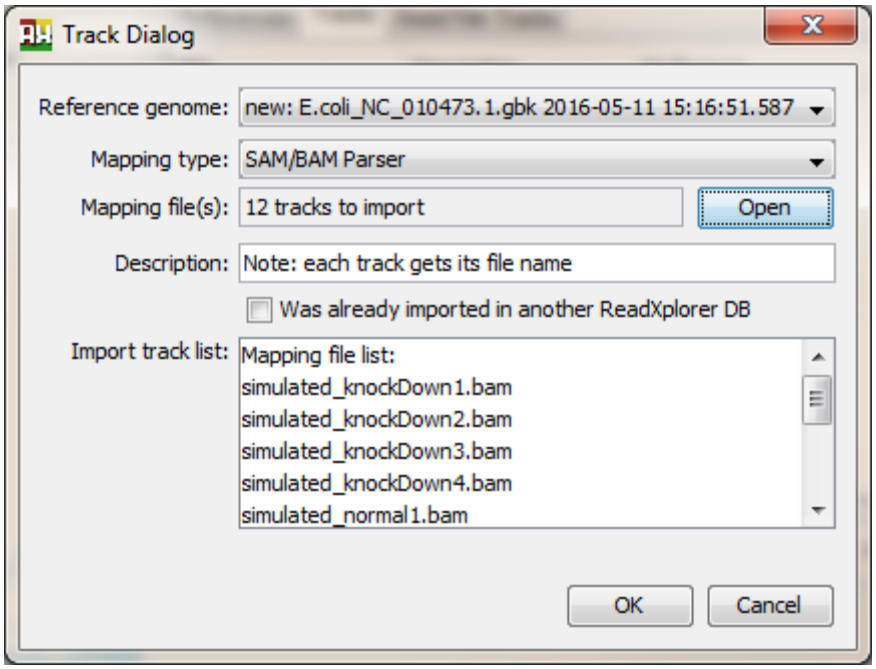
To create a new database within ReadXplorer go to “File” -> “Open/Create database” and choose the location and the name of your new DB. It usually makes sense to place the reference, the mapping files and the database in the same folder. ReadXplorer will tell you that the database does not exist and automatically creates a new one.



The newly created database is completely empty. To import your data go to “File” -> “Import Data”. At first you must import your reference. While in the “Reference” tab select “Add” and choose your reference file in any supported file format.



Once you have done that switch to the “Tracks” or “Read pair Tracks” tab (depending if you want to import single end or paired end data) and select “Add”. Navigate to your mapping files; you can select multiple files at once. Check if the right “Reference genome” is selected.



Once you have entered all data you want to import you can start the import process by clicking the “Finish” Button.

ReadXplorer will import the reference first. A Fasta file containing the sequence will be created and indexed. Once the reference is imported ReadXplorer will start to process the mapping files one after the other. When the input is in sam format it will be converted to bam. ReadXplorer classifies all the reads in the file and writes the result to a new bam file which can be identified by “_extended” at the end of the file name previous to the “.bam” file ending. A bam index file is also created for each bam file. Once the import is finished you can start using your database.

8. How can I increase the RAM limit?

As ReadXplorer is a Java application a RAM (heap space) limit must be specified before starting the program. This is done in the settings file “readxplorer.conf” that is located in the “etc” directory within the program directory. In this file you will find the parameter “-J-Xmx2G”, limiting the maximum amount of RAM Java can allocate to 2 gigabyte (GB). You can increase this limit by changing the “2” accordingly. You can also specify megabytes (MB) as follows: “-J-Xmx2500m”. The default limit of 2 GB should be sufficient for most use cases. However, if you import a lot of reads and ReadXplorer becomes unresponsive for a really long time or prints an “Out of memory”-error you should increase the limit.

9. I am running out of space, can I delete some of the mapping files?

ReadXplorer will keep the original sam/bam files untouched and only work on its own "extended" copies of these files. The ReadXplorer bam files all have the word “extended” in their name (e.g. simulated_upRegulated4_extended.bam) and are accompanied by a matching bam index file (e.g. simulated_upRegulated4_extended.bam.bai). Note that only mapped reads are kept in the extended mapping files. Unmapped reads are not stored to save some disk space. For the mapped reads, all original data is present in the extended bam files plus some flags for the read mapping classification carried out during the import into ReadXplorer. So if you do not need unmapped reads, you can safely delete your original sam/bam files after successfully importing them into ReadXplorer without losing any needed information. If you still want to delete the original files, but keep the unmapped reads, you can filter them into a separate file using samtools[6].

10. No read data is shown!

This is most likely a naming problem. In a sam/bam file it is noted down for each single read to which reference it belongs. If these entry does not match the name of the reference the track was assigned to during import, no data will be shown. All references used within the mapping file can be viewed with samtools[3] by inspecting the so called “sequence dictionary”. We are facing this problem quite often and hence there are some checks implemented during import that should warn you in most cases if you are trying to work with non-matching entries. A common source for this error is usually the usage of a Fasta and a GenBank file with non-matching reference names. The GenBank file is imported into ReadXplorer but as most mapping tools need the reference in Fasta format the non-matching Fasta file is used for this task. We have seen a lot of cases were, even if the underlying sequence is exactly the same, the reference name in GenBank and Fasta file differ. The safest and recommended way is to use the combination of Fasta and GFF3. Using this file type combination you can be sure that you are working with the same reference sequence at all times. If you must use GenBank check if the corresponding Fasta file is really exactly the same and if the names of the reference sequence in both files are also the same. Keep in mind that not all tools handle Fasta headers in a consistent way (it is e.g. a bad idea to have whitespaces in the Fasta header as some tools might truncate the header at the first whitespace while others don’t). If you want to be 100 % sure that GenBank and Fasta match there is a simple trick tough: import the GenBank file as a reference into ReadXplorer prior to mapping your read data. ReadXplorer will create a Fasta file with correct Fasta header during the import process. It is located in the same folder as the original GenBank file. If you use a copy of this file for the mapping process you should not have any problem with reference naming when importing the mapping data into ReadXplorer.

[6] http://samtools.sourceforge.net/