SyntenyViz Analysis Workflow

Contents

The course below demonstrates a standard analysis pipeline with the `SyntenyViz` R package.

For the inpatients

Quick and minimum steps to get start a synteney conservation anaysis with SyntenyViz

- Define an investigation range We need to firstly define an investigation range to cover the target range in gene coordinate. We will use a mouse dipeptidyl dipeptidase 4 gene (DPP4-mm) in this example, where DPP4-mm locates at chromosome number 2 between 62,330,073-62,412,231 bp.

```
# orgm is a handle for organism
orgmName <- "Mmusculus"
# mycoords.list is the investigation range handler
mycoords <- "2:6.0e7:6.5e7"
```

- Convert `mycoords.list` into a GRange object

```
mycoords.gr <- SyntenyViz::coordFormat (mycoords.list = mycoords)
```

It is always a good habit to double check the input, so

```
mycoords.gr
#> GRanges object with 1 range and 0 metadata columns:
#>       seqnames            ranges strand
#>          <Rle>         <IRanges>  <Rle>
#>   [1]     chr2 60000000-65000000      *
#>   -------
#>   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

- Construct a single synteny graph

```
synvizPlot(mycoords.gr, orgmName)
```

- Construct a multi synteny graph

Pick a few of targets

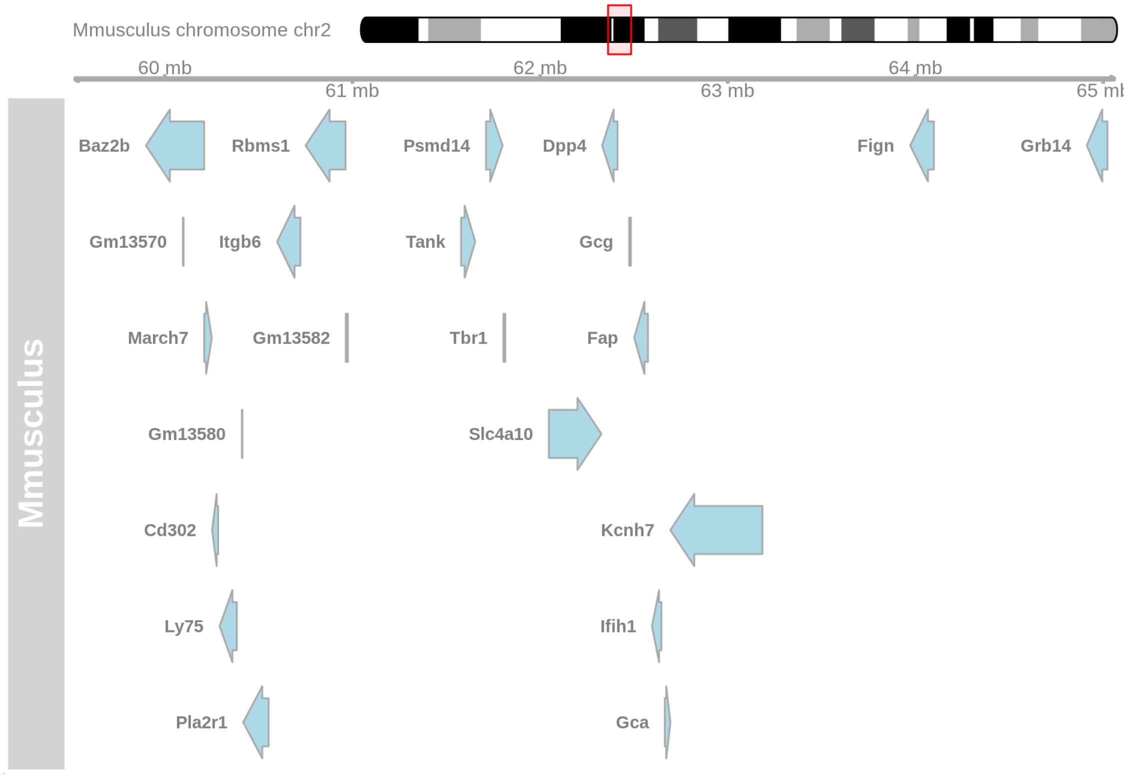

Figure 1: Synteny of Mouse DPP4

```
orgm.1 <- "Hsapiens"
mycoords.list.1 <- "2:15.95e7:16.45e7"
orgm.2 <- "Mmusculus"
mycoords.list.2 <- "2:6.0e7:6.5e7"
orgm.3 <- "Rnorvegicus"
mycoords.list.3 <- "3:4.6e7:5.1e7"
```

Then construct a multiple synteny query

```
orgmsList <- orgmsCollection.init (orgmsList)
orgmsList <- orgmsAdd (orgm.1, orgmTxDB, mycoords.list.1, orgmsList)
orgmsList <- orgmsAdd (orgm.2, orgmTxDB, mycoords.list.2, orgmsList)
orgmsList <- orgmsAdd (orgm.3, orgmTxDB, mycoords.list.3, orgmsList)
```

Now, construct a comparative multi-synteny graph

```
multiplot <- multisynvizPlots(orgmsList)
```

The Nitty Gritty of Synteny Analysis

Construct A Search Query

A search query includes the information of target organism(s) and the region(s) of interests. A typical work flow starts by locating the gene of interests. In this example, we will use a protease gene dipeptidyl dipeptidase-4 (DPP4) as a target.

DPP4 background

DPP4 is also known by a few other names including adenosine deaminase complexing protein 2 (ADCP2) and CD26. Human DPP4 gene encodes a 766 amino acids protein. DPP4 proteins dimmerise and then possesses potent rare post-proline hydrolytic activity *in situ*. In clinics, DPP4 has been demonstrated to involve in many important physiobiological functions including homeostasis, adaptive immune responses and numerous cancer biology. Due to its degradation property towards GLP and GIP, pharmacologically, DPP4 inhibitors have been prescribed as an alternative treatment to type 2 diabetics.

```
knitr::include_graphics('images/DPP4.png')
```

Locate DPP4 gene and define the investigation ranges

Locate gene (e.g. DPP4 gene)

To locate DPP4 gene, both Ensemble and GeneBank are good resources for this purpose.

Let's say we want to find the location of **human DPP4 gene** location.

- Ensemble query
- GeneBank query

indicate human DPP4 gene is located on **chromosome 2** between **161992245-162074215 bp**.

Define the investigation ranges

Given that human DPP4 related genes are clustered closely and we only interested in the synteny of DPP4 gene in this case, therefore, **0.25e7 bp** on either side of DPP4 gene seems a good fit (i.e. **15.95e7-16.45e7** bp will be the investigation range).

Hence, we can now define the investigation range as

```
# orgm is a handle for organism
orgm <- "Hsapiens"
# mycoords.list is the investigation range handler
mycoords.list <- "2:15.95e7:16.45e7"
```

then we need to convert `mycoords.list` into a squence coordinate in GRange object, here we can call a convertion function

```
# Return mycoords.gr as a GRange object
mycoords.gr <- coordFormat (mycoords.list = mycoords.list)
```

Prepare the environment for analysis

SyntenyViz v0.0.0.9000 utilises UCSC transcriptomics database in TxDB object format. Available analysis can be performed on following databases.

Table 1: Available transcriptomics dataset for analysis as in SyntenyViz v0.0.0.9000

dbClass	dbSpecies	dbSource	dbAbbv	dbType
TxDb	Btaurus	UCSC	bosTau9	refGene
TxDb	Celegans	UCSC	ce11	refGene
TxDb	Cfamiliaris	UCSC	canFam3	refGene
TxDb	Dmelanogaster	UCSC	dm6	ensGene

dbClass	dbSpecies	dbSource	dbAbbv	dbType
TxDb	Drerio	UCSC	danRer11	refGene
TxDb	Ggallus	UCSC	galGal6	refGene
TxDb	Hsapiens	UCSC	hg38	knownGene
TxDb	Mmulatta	UCSC	rheMac10	refGene
TxDb	Mmusculus	UCSC	mm10	knownGene
TxDb	Ptroglodytes	UCSC	panTro6	refGene
TxDb	Rnorvegicus	UCSC	rn6	refGene
TxDb	Scerevisiae	UCSC	sacCer3	sgdGene
TxDb	Sscrofa	UCSC	susScr11	refGene

Organisms can be analysised as in v0.0.0.9000 include

Table 2: Available organisms for analysis as in SyntenyViz v0.0.0.9000

dbClass	dbSpecies	dbSource	dbType
org	Hs	eg	db
org	Mm	eg	db
org	Rn	eg	db
org	Sc	sgd	db
org	Dm	eg	db
org	At	tair	db
org	Dr	eg	db
org	Ce	eg	db
org	Bt	eg	db
org	Gg	eg	db
org	Cf	eg	db
org	Ss	eg	db
org	Mmu	eg	db
org	EcK12	eg	db
org	Xl	eg	db
org	Ag	eg	db
org	Pt	eg	db
org	Pf	plasmo	db
org	EcSakai	eg	db

A translation on the organism abbriviations may be useful here, so

Table 3: Reference Card for organisms Names

Scientific.Names	Common.Names	Abbrav	Abbrav.Short
Bos taurus	Cow	Btaurus	Bt
Caenorhabditis elegans	Roundworm	Celegans	Ce
Canis familiaris	Dog	Cfamiliaris	Cf
Drosophila melanogaster	Fruitfly	Dmelanogaster	Dm
Danio rerio	Zebrafish	Drerio	Dr
Gallus gallus	Chicken	Ggallus	Gg
Homo Sapiens	Human	Hsapiens	Hs
Macaca mulatta	Rhesus macaque	Mmulatta	Mmu
Mus musculus	House mouse	Mmusculus	Mm

Scientific.Names	Common.Names	Abbrav	Abbrav.Short
Pan troglodytes	Chimpanzee	Ptroglodytes	Pt
Rattus norvegicus	Brown rat	Rnorvegicus	Rn
Saccharomyces cerevisiae	Brewer	Scerevisiae	Sc
Sus scrofa	Wild swine	Sscrofa	Ss

Loading the orgnism databases for further analysis

```
orgm_OrgDB <- getPkgs(orgm, orgmOrgDB)
orgm_TxDB <- getPkgs(orgm, orgmTxDB)
```

Generate a dataset for a single Synteny Plot Construction

Function `synvizPlotData` can be used to automatically generate `Gviz` tracks for synteny plotting.

```
plotData <- synvizPlotData(mycoords.gr =  mycoords.gr, orgm = orgm)
```

Here is a good time to check the integraty of data, so let's have a look `plotData`

```
plotData
#> $itrack
#> Ideogram track 'Hsapiens chromosome chr2' for chromosome 2 of the hg38 genome
#>
#> $gtrack
#> Genome axis 'Axis'
#>
#> $atrack
#> AnnotationTrack 'Hsapiens'
#> | genome: hg38
#> | active chromosome: chr2
#> | annotation features: 29
```

Generate a single Synteny Plot

We will now use function `syntenyPlot` to construct a single synteny plot.

```
synvizPlot (mycoords.gr = mycoords.gr, orgm = orgm)
```

Generate a Multiple Synteny Plot

In evolution biology, the conservation in synteny across species reveals important evolution tracks. So, let's have a look how DPP4 synteny conserved between human, mouse and rat.

```
orgm.1 <- "Hsapiens"
mycoords.list.1 <- "2:15.95e7:16.45e7"
orgm.2 <- "Mmusculus"
mycoords.list.2 <- "2:6.0e7:6.5e7"
orgm.3 <- "Rnorvegicus"
mycoords.list.3 <- "3:4.6e7:5.1e7"

orgmsList <- orgmsCollection.init (orgmsList)
orgmsList <- orgmsAdd (orgm.1, orgmTxDB, mycoords.list.1, orgmsList)
orgmsList <- orgmsAdd (orgm.2, orgmTxDB, mycoords.list.2, orgmsList)
orgmsList <- orgmsAdd (orgm.3, orgmTxDB, mycoords.list.3, orgmsList)
```

We now can call `synvizPlot` three times, or we can use function `multiplot` to do this for us.

```
multiplot <- multisynvizPlots(orgmsList)
```

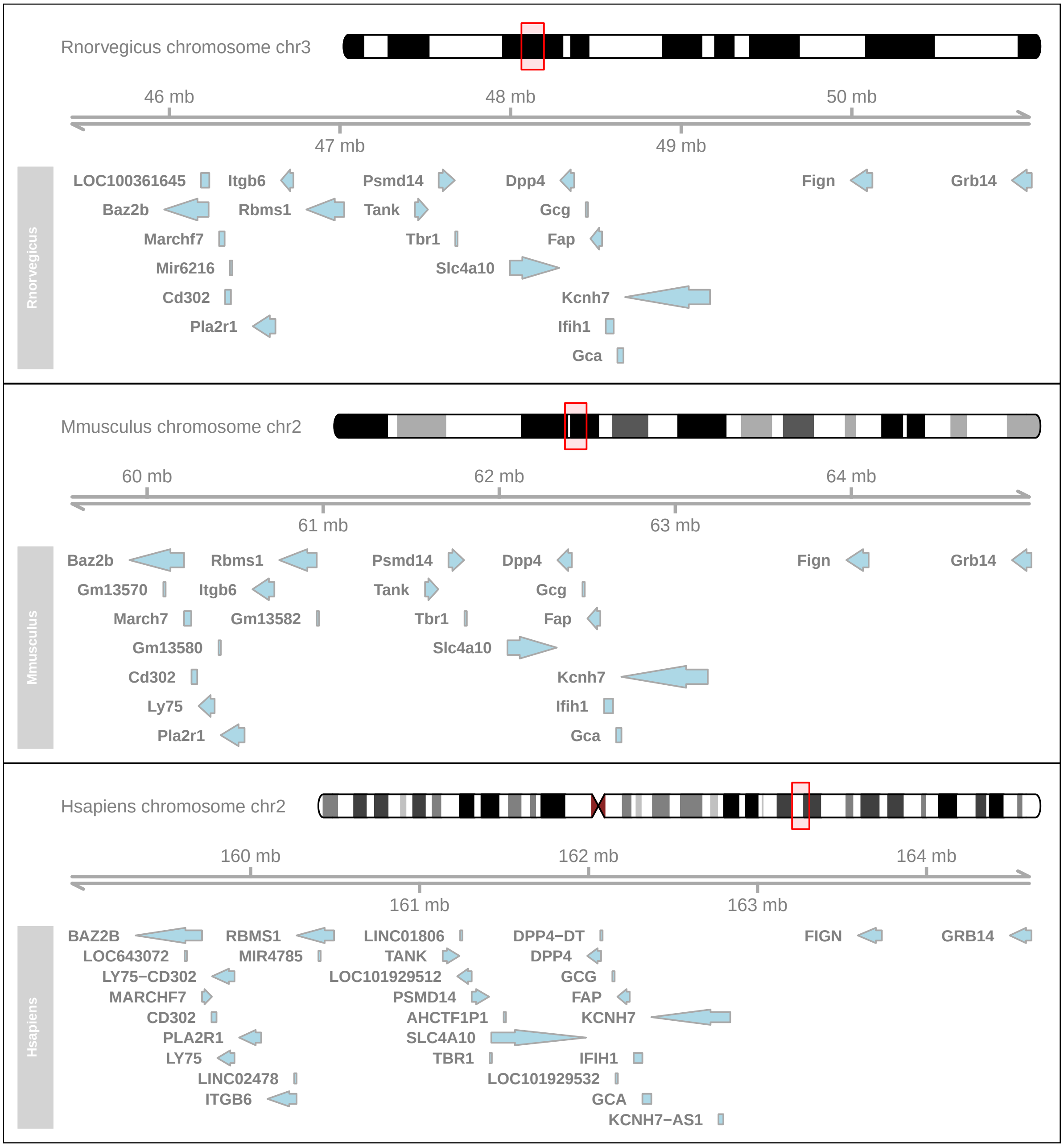

Figure 2: Multi Synteny Plot of DPP4

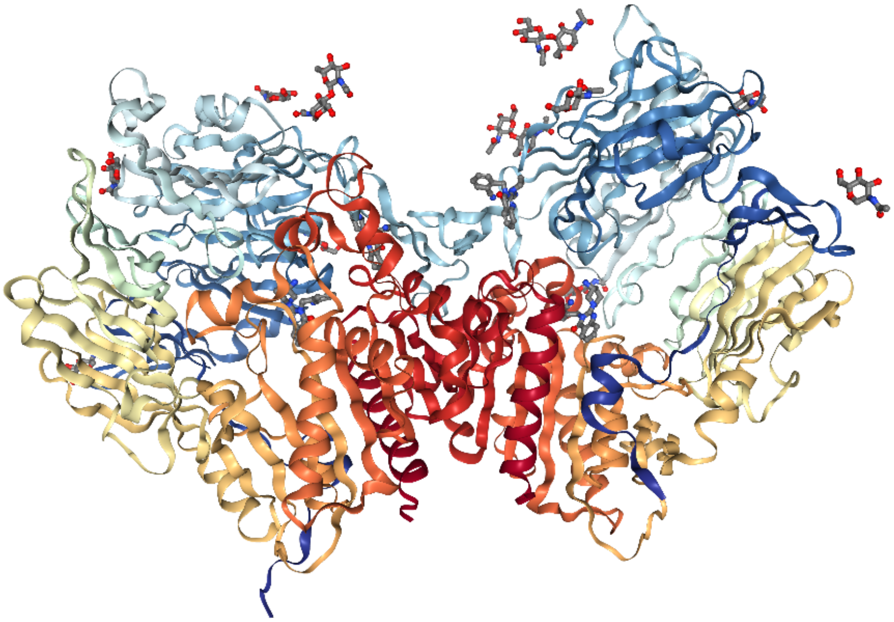

Figure 3: DPP4 protein in dimmer form

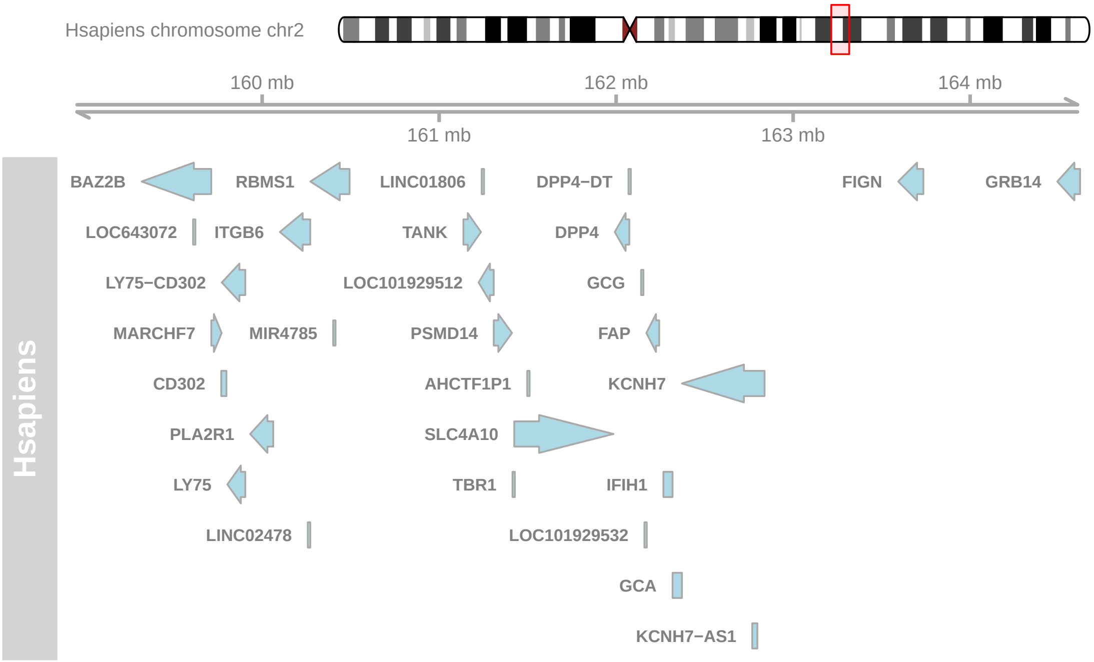

Figure 4: Synteny of Mouse DPP4

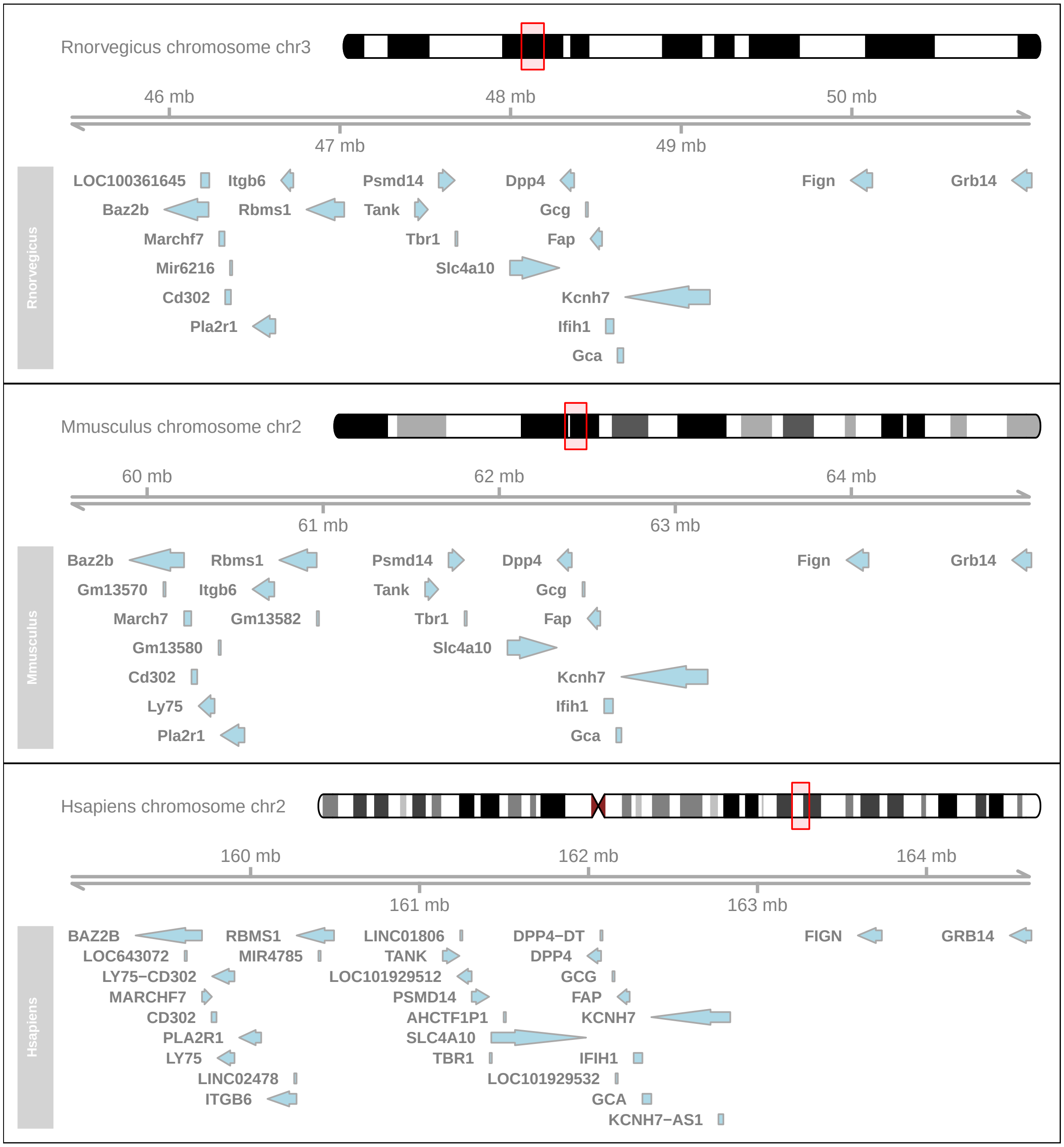

Figure 5: Multi Synteny Plot of DPP4