Institute for Advanced Study

One at a time or all at once: Simulating the formation of Kepler systems

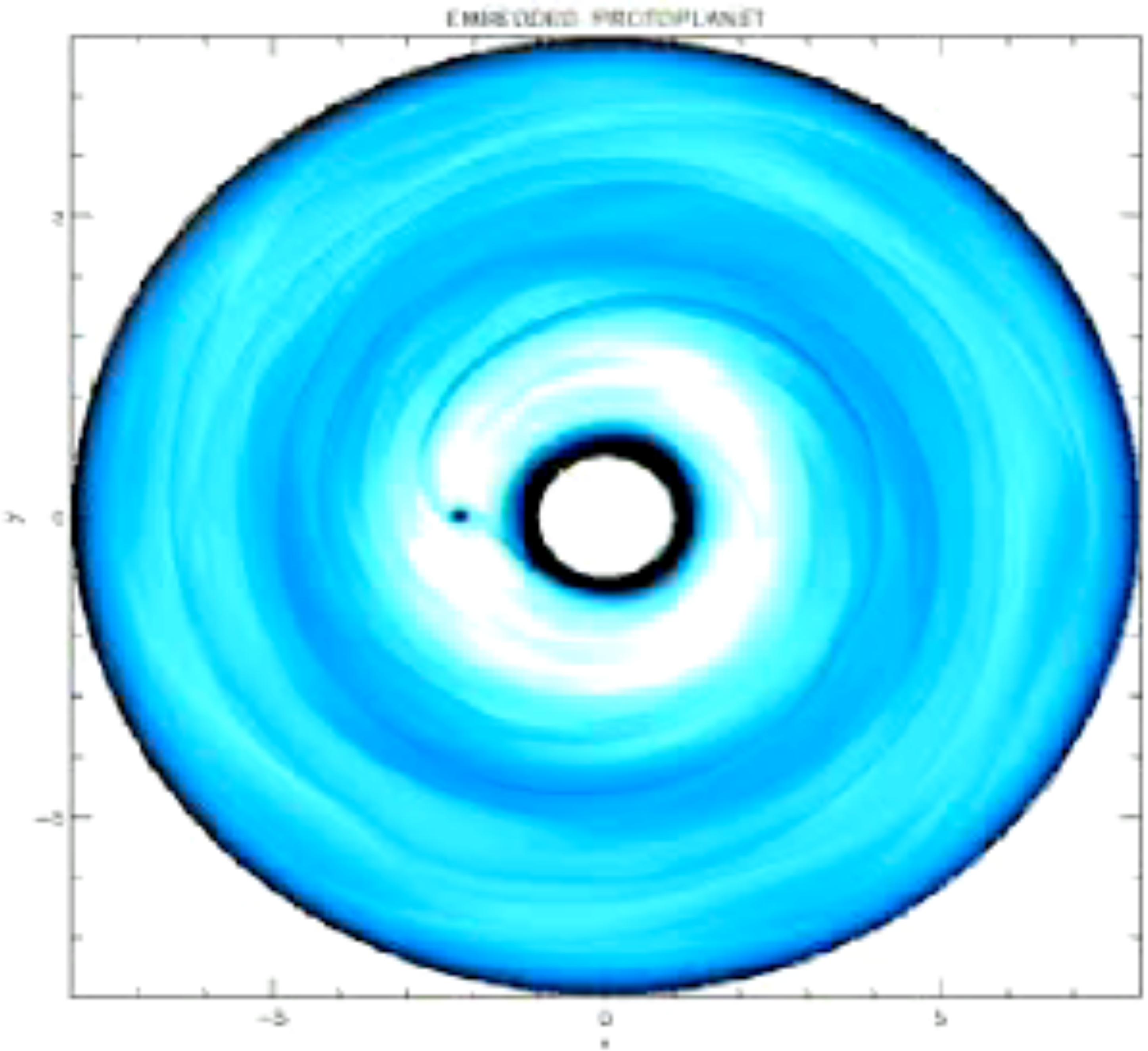
Hanno Rein

Evolution of planets, the simplest case

1. Planets form in a protoplanetary disk
2. Planets form beyond the snowline
3. Planets interact with the disk in which they formed

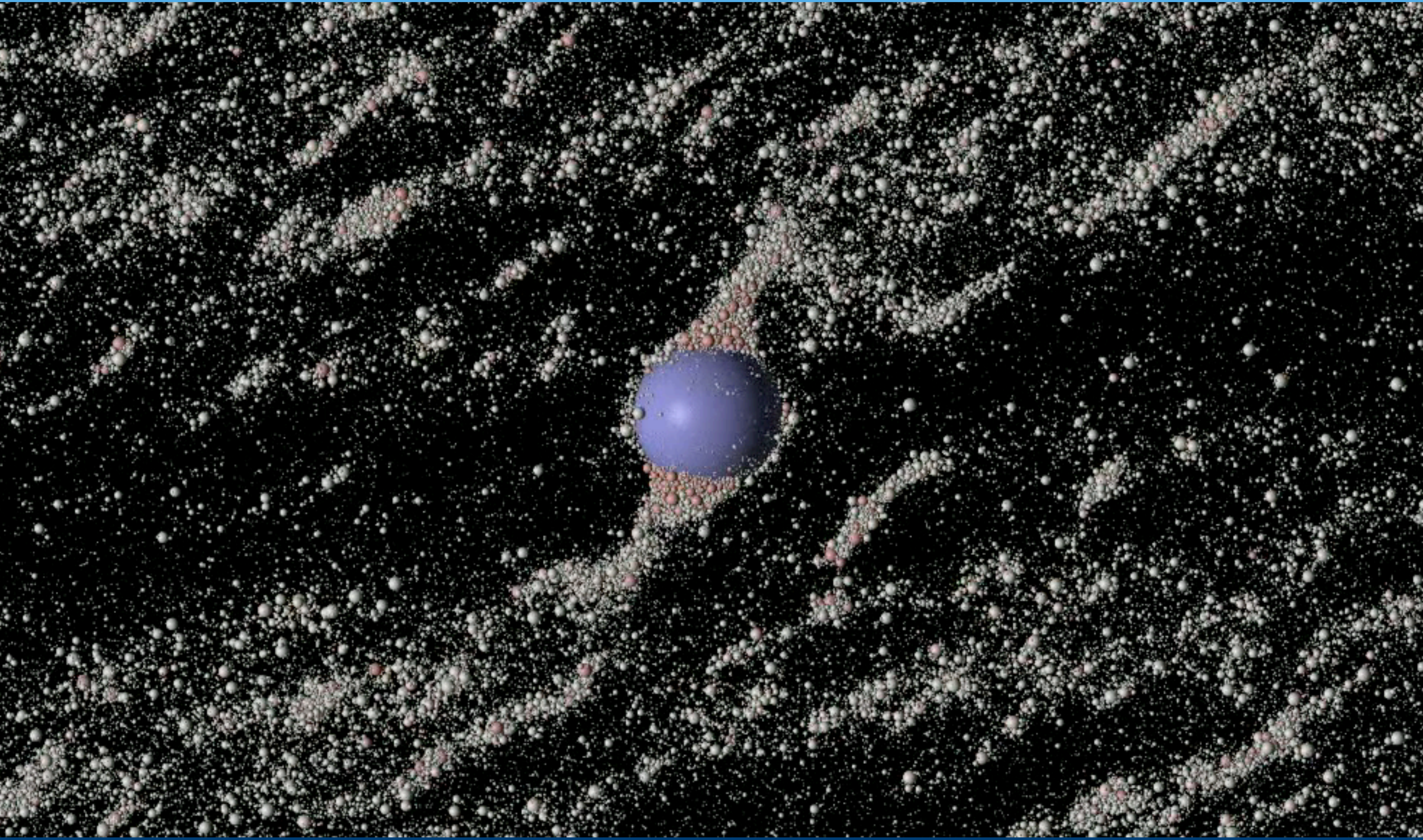
Turbulent disk

- Measured angular momentum transport requires some sort of turbulent transport
- Magnetorotational instability (MRI)
- Stochastic forces lead to random walk of the planet



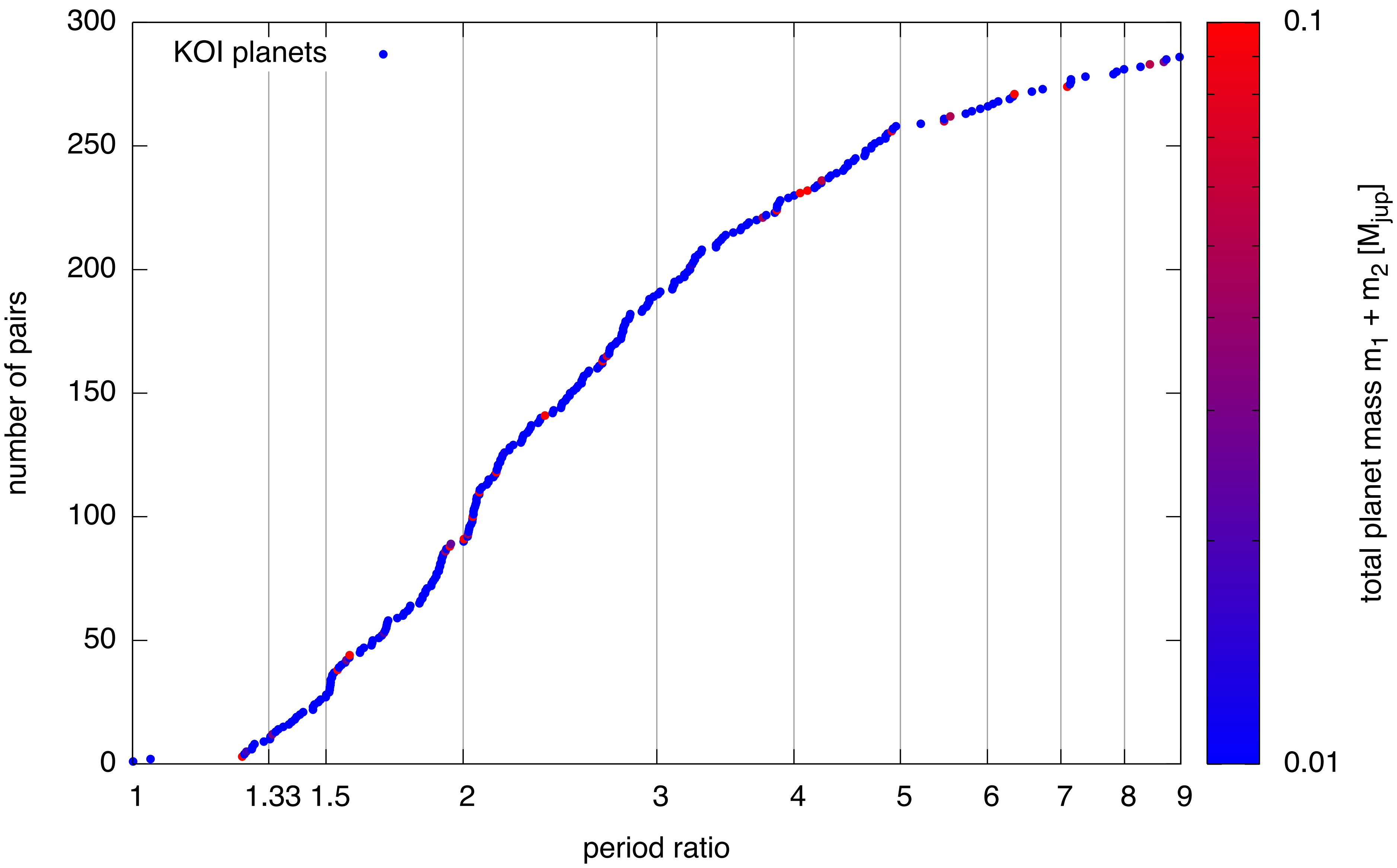
Animation from Nelson & Papaloizou 2004
Laughlin et al. 2004, Nelson 2005, Oischi et al. 2007, Rein & Papaloizou 2009

Random walk in Saturn's Rings: Propellers



REBOUND code, Rein & Papaloizou 2010, Crida et al 2010

Kepler's transiting planet candidates



- Period ratio distribution much smoother for small mass planets
- Deficiencies near 4:3, 3:2, 2:1
- Excess slightly outside of the exact commensurability

Rein, Payne, Veras & Ford (2012)

Method

Architecture and masses from observed KOIs

Placing planets in a MMSN, further out, further apart, randomizing all angles

N-body simulation with migration forces

Advantages

Comparison of statistical quantities

- Period ratio distribution
- Eccentricity distribution
- TTVs

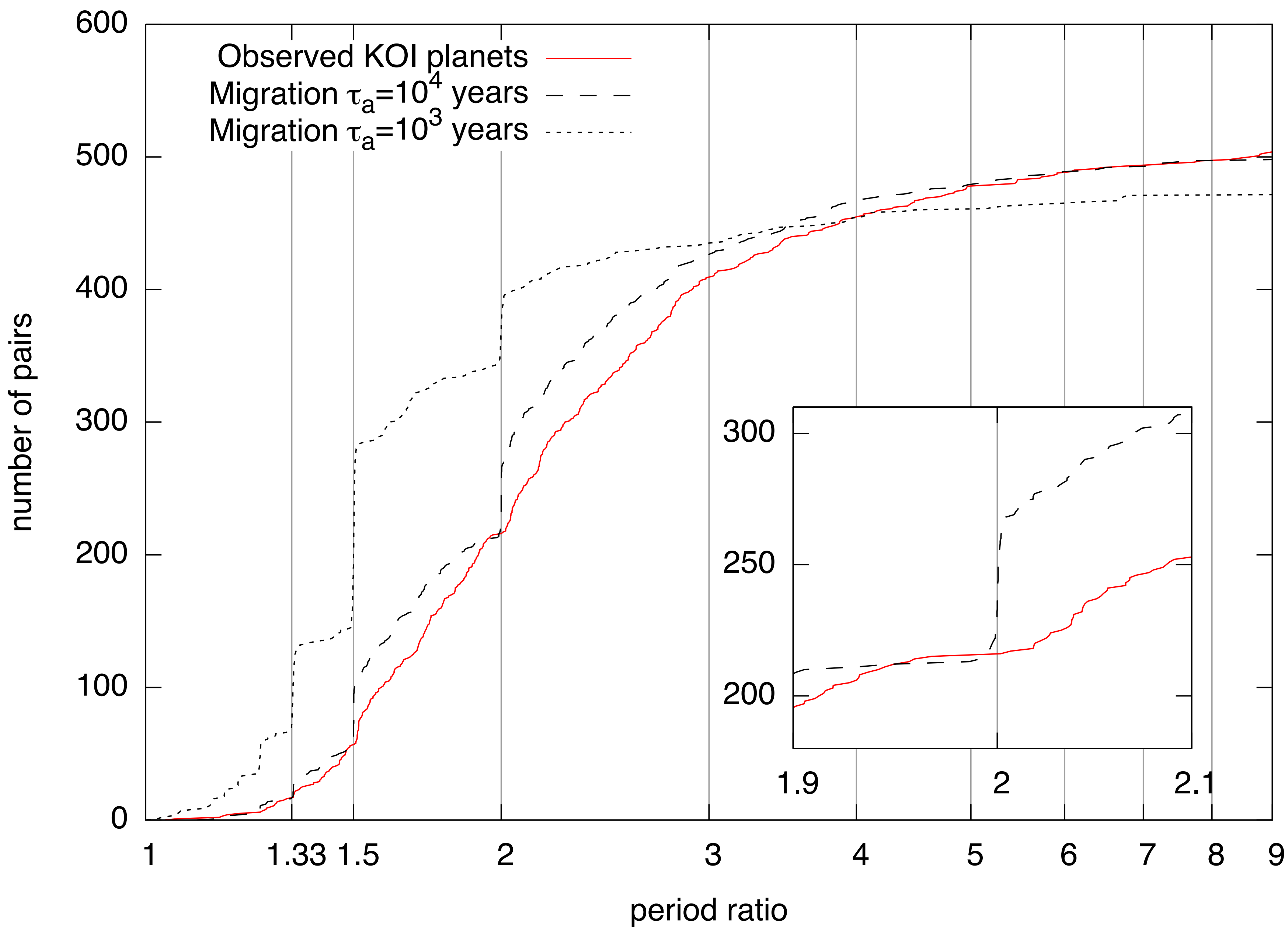
Comparison of individual systems

- Especially interesting for multi-planetary systems
- Can create multiple realizations of each system

No synthesis of a planet population required

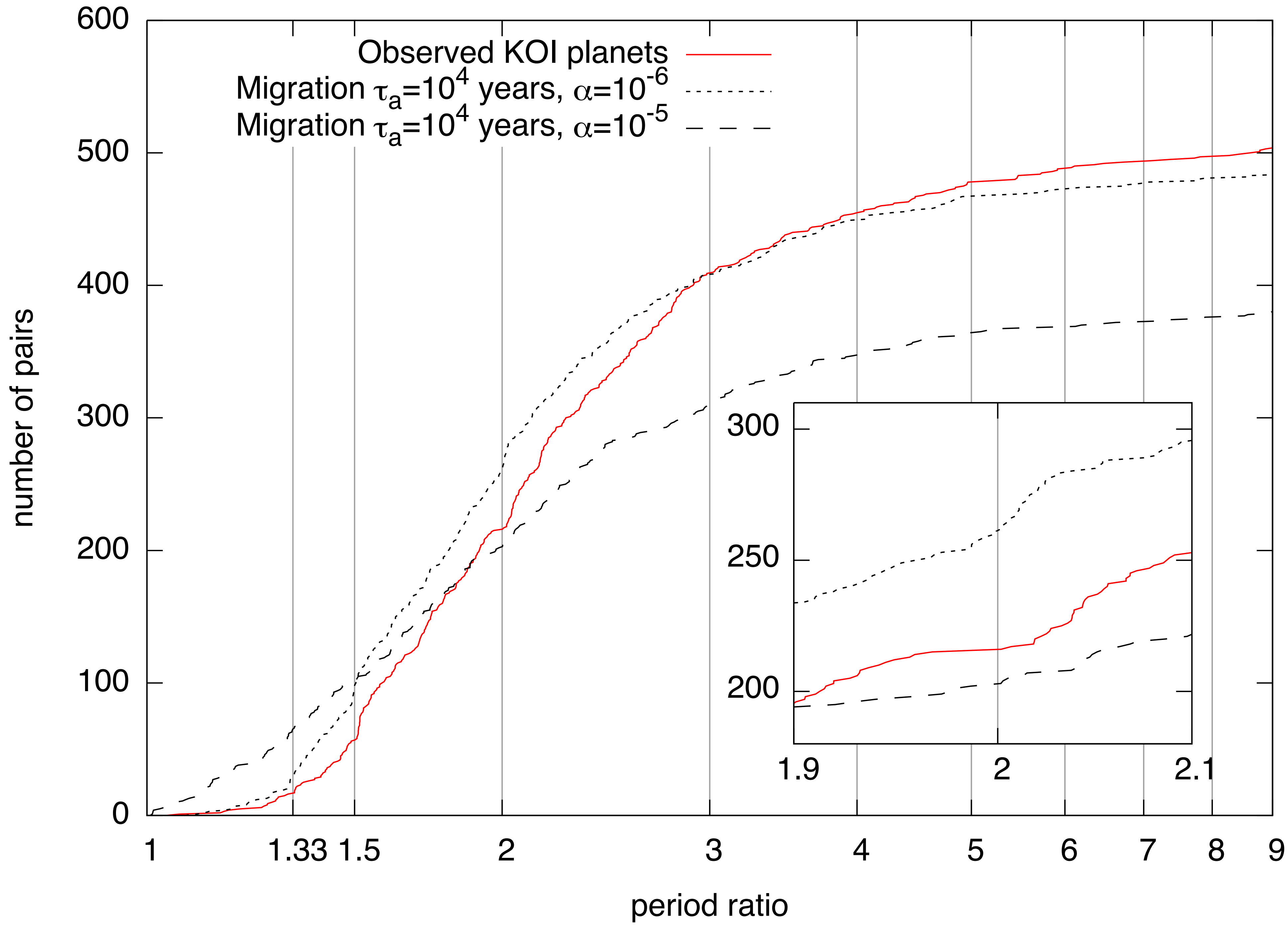
- Observed masses
- Observed architectures

Result 1: Smooth migration alone is not enough



Rein 2012

Result II: Stochastic migration works much better



Conclusions

1. Simplest scenario seems to work without fine-tuning.
2. This method can easily be extended to study other ideas such as stellar tides, secular chaos, etc.
3. This method allows the simultaneous comparison of individual and ensemble quantities.

Institute for Advanced Study

Open Exoplanet Catalogue

Hanno Rein

Why do we need another exoplanet catalogue?

Centralized

- Impossible to correct typos, add data without sending an e-mail to the person in charge
- Closed ecosystem

Slow and outdated

- It can take days/weeks/months for new planets to be added
- Maintainer can be holiday or abandon the project

Web-based

- Website are badly written
- Requires flash or java plugin
- Need a constant internet connection
- Restricted to a very limited, predefined set of possible queries

Old-fashioned formats

- Static tables are not adequate to represent diverse dataset
- Almost impossible to include binary/triple/quadruple systems
- Not flexible when adding new data
- Unintuitive to parse

Open Exoplanet Catalogue

Demo

Example of a python script parsing all systems

```
import xml.etree.ElementTree as ET, glob
for filename in glob.glob("*.xml"):
  tree = ET.parse(open(filename, ’r’))
  planets = tree.findall(".//planet")
  for planet in planets:
    print planet.findtext("./name")
    print planet.findtext("./mass")
```

Open Exoplanet Catalogue

Open source philosophy

- Unrestrictive MIT license
- Community project
- Everyone can contribute and modify data
- Everyone can expand it
- Distributed, no need for a server/website
- Private clones with confidential data

Based on git

- Distributed version control system
- Used by Linux kernel and most other open source projects
- Every single value, every change ever made is logged, verifiable

Ready to go

- 674 systems, 51 binary system, 870 exoplanets, 9 solar system objects, 2740 KOI objects
- ~10 million users

Hierarchical data structure

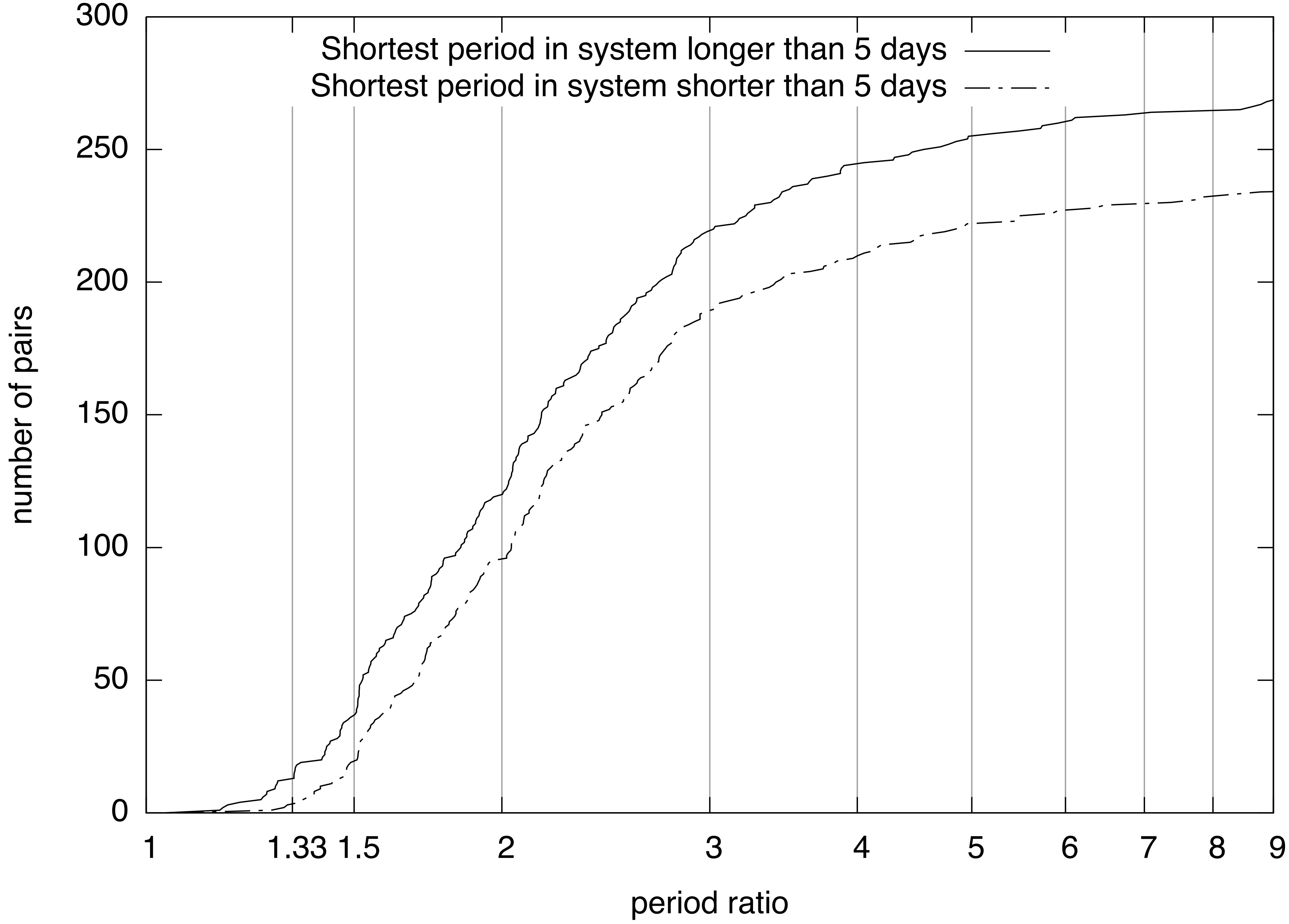
- Uses plain XML
- Can represent arbitrary configurations in systems with stellar multiplicity >1
- Extremely easy and intuitive to parse in almost any language
- Compresses extremely well
- size ~ 100KB

OpenExoplanetCatalogue.com

arXiv:1211.7121

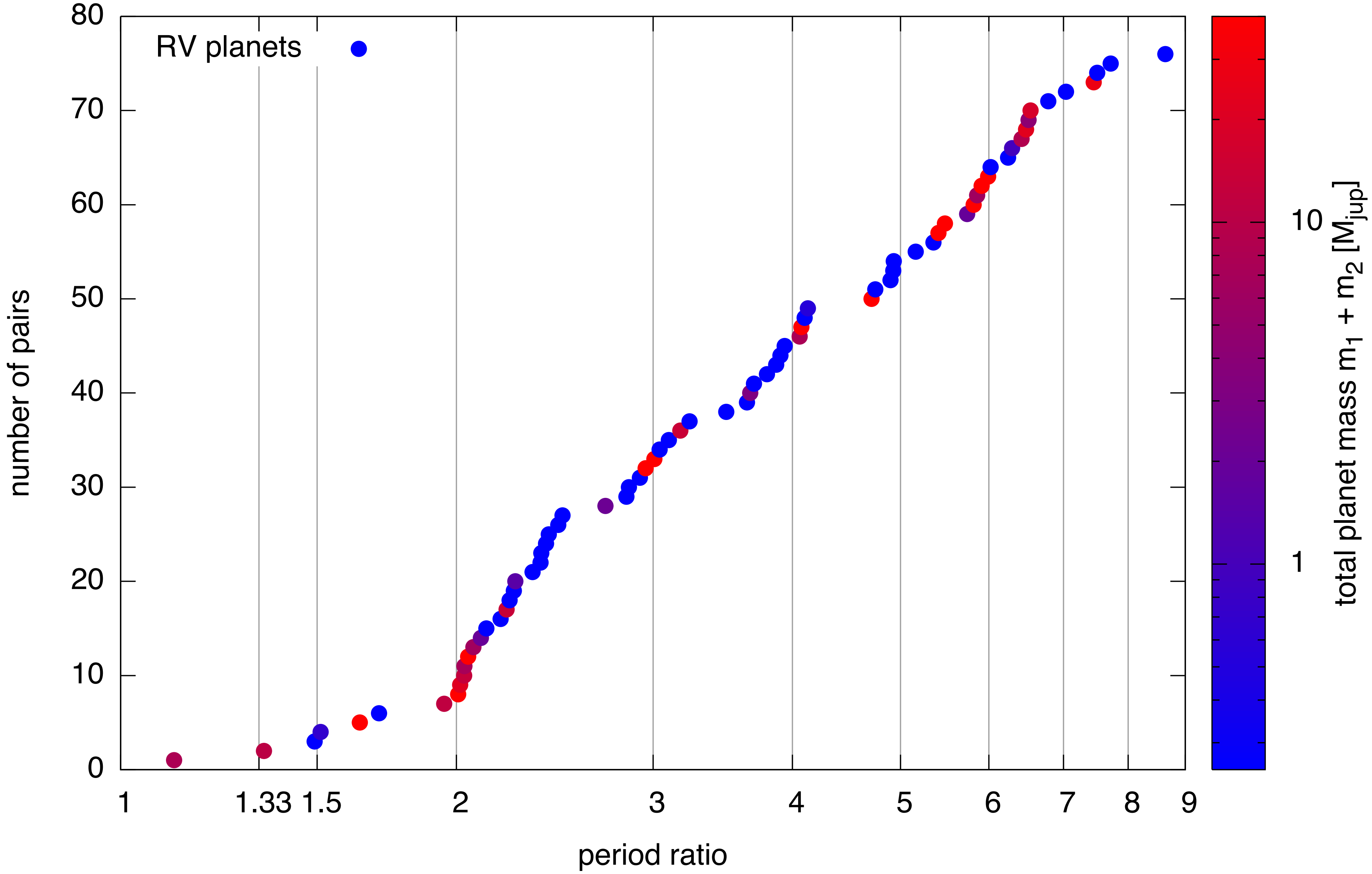
Backup slides

No different for close-in/far-out planets



Rein 2012

Radial velocity planets



- Periods of systems with massive planets tend to pile up near integer ratios
- Most prominent features at 4:1, 3:1, 2:1, 3:2

Example of a system file: 42 Dra b

```xml
<system>
    <name>42 Dra</name>
    <rightascension>18 25 59</rightascension>
    <declination>+65 33 49</declination>
    <distance>97.3</distance>
    <star>
        <mass>0.98</mass>
        <radius>22.03</radius>
        <magV>4.83</magV>
        <metallicity>-0.46</metallicity>
        <spectraltype>K1.5III</spectraltype>
        <planet>
            <name>42 Dra b</name>
            <list>Confirmed planets</list>
            <mass>3.88</mass>
            <period>479.1</period>
            <semimajoraxis>1.19</semimajoraxis>
            <eccentricity>0.38</eccentricity>
            <description>42 Draconis is a metal poor star.</description>
            <discoverymethod>RV</discoverymethod>
            <lastupdate>09/03/23</lastupdate>
            <discoveryyear>2009</discoveryyear>
            <new>0</new>
        </planet>
        <name>42 Dra</name>
    </star>
</system>
```

Example of an HTML site parsing all systems

```
<html>
    <head>
        <script type="text/javascript" src="./js/d3.v3.min.js"></script>
        <script type="text/javascript" src="./js/jquery.min.js"></script>
    </head>
    <body>
        <table>
            <thead><tr><th>Name</th><th>Mass [MJup]</th></tr></thead>
            <tbody id="tablebody"></tbody>
        </table>
        <script type="text/javascript">
            d3.xml("systems.xml", "application/xml", function(xmldata) {
                var planets = $("planet",xmldata);
                var tr = d3.select("#tablebody").selectAll("tr")
                    .data(planets).enter().append("tr");
                tr.append("td").text(function(planet){
                        return $("name:first",$(planet)).text();
                    });
                tr.append("td").text(function(planet){
                        return $("mass",$(planet)).text();
                });
            });
        </script>
    </body>
</html>
```